Ref. Ares(2017)3234053 - 28/06/2017

Grant Agreement No. 619572

COSIGN

Combining Optics and SDN In next Generation data centre Networks

Programme: Information and Communication Technologies

Funding scheme: Collaborative Project – Large-Scale Integrating Project

Deliverable D5.4 – COSIGN and Future DCN Architecture Requirements

Due date of deliverable: 31st March, 2017
Actual submission date: 9th June, 2017

Start date of project: January 1, 2014 Duration: 39 months

Lead contractor for this deliverable: i2CAT and DTU

Project co-funded by the European Commission within the Seventh Framework Programme		
Dissemination Level		
PU	Public	X
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Executive Summary

This document reports the implementation and outcome of the COSIGN demonstrators. A number of key achievements of the integration and demonstration efforts are summarized below.

We demonstrate an improvement of up to 42% in the number of accepted VDC instances on the same infrastructure when performing our developed joint provisioning of VMs and network configuration as opposed to legacy approaches with independent configuration.

We demonstrate 150% improvement in throughput for selected elephant flows by applying the vApp traffic monitoring and data plane reconfiguration in an advanced circuit sharing scheme. The 150% improvement comes at the cost of only 37.5% reduction in throughput for the existing flow on the shared connection. The improvement from optical circuit reconfiguration increases to even higher values when higher interface rates are employed towards the optical switch – which is agnostic to this change.

We demonstrate 50% improvement in round trip time (RTT) for mice flows when using circuit sharing according to the vApp use case.

We have confirmed in an industrial setting that the COSIGN VDC approach introduces negligible delay and complexity on the network side making it a highly attractive approach for industrial operators to investigate further.

We have provided a significant advance in the automated provisioning of VDCs as the COSIGN VDC approach can provide VDCs with guaranteed QoS with the same ease and speed on the part of the provider as "best-effort" VDCs are provided today. This is also confirmed in the industrial demonstrator and will potentially provide great value for operators adopting this approach.

We have conducted a proof of concept installation of a Hypercube structure with an optical shortcut switch and we furthermore present a framework for simulation and real world device integration that allows for performance evaluation and scalability studies of multiple of datacentre architectures.

The COSIGN demonstration and integration efforts have been completed with great success. The project has disseminated knowledge and results to various audiences and has produced significant scientific progress in the field of datacentre networks.

Legal Notice

The information in this document is subject to change without notice.

The Members of the COSIGN Consortium make no warranty of any kind with regard to this document, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The Members of the COSIGN Consortium shall not be held liable for errors contained herein or direct, indirect, special, incidental or consequential damages in connection with the furnishing, performance, or use of this material.

Possible inaccuracies of information are under the responsibility of the project. This report reflects solely the views of its authors. The European Commission is not liable for any use that may be made of the information contained therein.

Document Information

Status and Version:	Final version	
Date of Issue:	09/06/17	
Dissemination level:	Public	
Author(s):	**Name**	**Partner**
	José Aznar	I2CAT
	Albert Vinyes	I2CAT
	Daniel Guija	I2CAT
	Domenico Gallico	IRT
	Matteo Biancani	IRT
	Marco Capitani	NXW
	Chris Jackson	UNIVBRIS
	Yaniv Ben Itzhak	IBM
	Albert Pagès	UPC
	Artur Pilimon	DTU
	Sarah Ruepp	DTU
	Michael Galili	DTU
	Jakob Thrane	DTU
Edited by:	José Aznar	I2CAT
	Michael Galili	DTU
Reviewed by :	Michael Enrico	Polatis
	Hans Christian Hansen Mulvad	UNISOUTH
Checked by:	Sarah Ruepp	DTU

Table of Contents

1 Introduction

This document constitutes the final WP5 deliverable and reports on the most significant outcomes of the COSIGN project mid-term and long-term demonstrators that were proposed in the demonstration plan of the previous deliverable D5.3 [3]. It also includes concrete performance metrics, results and the analysis of features. These demonstrators are aligned with the main COSIGN objective: to advance towards future datacentre networks where highly efficient optical interconnect technologies, controlled by the advanced programmable software control plane, satisfy the diverse and dynamic requirements of modern datacentre workloads as it was stated in the Description of Work (DOW) document [1].

In particular, the following demonstrators were planned: the mid-term demonstrator that was shown at the ECOC conference in Dusseldorf (September 2016), the industrial mid-term scenario validation hosted in IRT premises, and two final COSIGN demonstrators: a large-scale midterm demonstrator implementation hosted at DTU and the long-term scenario demonstrator hosted by University of Bristol. These demonstrators represent the COSIGN vision while addressing the challenging migration towards all optical DCNs, supported by the SDN and orchestration technologies. They clearly demonstrate the advantages brought by the project. For each of the demonstrators, we present the scenario, the implementation details and the main results.

It is also worth mentioning the importance of the demonstrators' validation from a three-fold perspective: firstly, the value of the different developed technologies in WP2, WP3 and WP4 as well as the innovation of integrating all of them towards the COSIGN main goal. Secondly, the validation of the architecture design proposed in WP1 (see D1.4 [14]) for the different scenarios integrating the aforementioned COSIGN technologies. Finally, the validation of the use case services and their requirements as well as the potential improvements for the VDC use case compared to the current service provisioning.

The 3 months project extension has signified an important opportunity to strengthen the integration and dissemination efforts in the project. Practically this has offered the opportunity to demonstrate the long-term scenario at OFC, to showcase the mid-term demonstrators to the general public as part of "Forskningens Døgn" (The Day of Research) in Denmark and finally to produce additional tests and measurements on the integrated demonstrators.

Despite a persistent effort done by Venture to realize the integrated fast optical cross-point switch (OXS) for use in the long-term scenario, this could not be achieved within the project. Consequently, the so-called 'plan-B' switch has been employed. This is a modified version of the switch developed in the LIGHTNESS project.

The remaining sections of this document are structured as follows: **Section 2** reports on the mid-term scenario demonstrated at ECOC. Besides the scenario it includes the test-plan, overall mid-term scenario results obtained while testing the performance of the VDC and vApp use cases. **Section 3** consists of the industrial validation of the VDC use case running on top of the mid-term scenario. IRT has deployed this scenario in their premises and tested concrete aspects of the VDC service performance. When possible, it has been compared to the current VDC service provided by IRT. **Section 4** explains the COSIGN final demonstrators: the large-scale validation of the mid-term scenario and the long-term cluster-based scenario. Finally, **section 5** concludes on the overall COSIGN demonstrator outcomes and summarizes the key achievements.

1.1 Reference Material

1.1.1 Reference Documents

[1]	COSIGN FP7 Collaborative Project Grant Agreement Annex I - "Description of Work"
[2]	COSIGN Deliverable D4.5 – Next Generation Data Centre Resource Orchestration with COSIGN DCN
[3]	COSIGN Deliverable D5.3 - Plan for Integration and Testing of COSIGN Demonstrators
[4]	COSIGN Deliverable D5.2 - Demonstrator Result of Data Plane and SDN Environment Integration
[5]	S. Ruepp, A.Pilimon, J. Thrane, M. Galili, M.Berger, and L. Dittmann. "Combining Hardware and Simulation for Datacenter Scaling Studies", in proc. ONDM May 2017
[6]	Riverbed System in the Loop Tool (SITL). www.riverbed.com (formerly known as OPNET Inc.)
[7]	D. Abts, J. Kim. "High Performance Datacenter Networks: Architectures, Algorithms, and Opportunities", in Synthesis Lectures on Computer Architecture (2011)
[8]	A. M. Fagertun, M. Berger, S. Ruepp, V. Kamchevska, M. Galili, L. K. Oxenløwe and L. Dittmann (2015). "Ring-based All-Optical Datacenter Networks", in Proc. of European Conference on Optical Communications (ECOC) , Valencia, Spain
[9]	J. Ho Ahn, N. Binkert, A. Davis, M. McLaren, R. S. Schreiber. "HyperX: Topology, Routing, and Packaging of Efficient Large-Scale Networks", in Proc. of the Conference on High Performance Computing Networking, Storage and Analysis - SC '09 (2009)
[10]	SITL session, Opnetwork conference 2010, www.opnetwork.com
[11]	Xena Networks, www.xenanetworks.com, Xena Bay L2-3 and Xena Scale L4-7 Testers
[12]	Microsoft, "Windows Sockets Error Codes," 2017. [Online]. Available: https://msdn.microsoft.com/en us/library/windows/desktop/ms740668
[13]	The Windows packet capture library, https://www.winpcap.org/.
[14]	COSIGN Deliverable D1.4 - Architecture Design
[15]	COSIGN Deliverable D5.0 - Definition, Design and Test Plan for Use Cases
[16]	COSIGN Deliverable D5.1 - Demonstrator Results of Data Plane Integration of Different Switch and Fibre Solutions
[17]	www.hyperglance.com

1.1.2 Acronyms and Abbreviations

Most frequently used acronyms in the Deliverable are listed below. Additional acronyms may be defined and used throughout the text.

DoW	Description of Work
SDN	Software Defined Networks
OCS	Optical Circuit Switch
OFC	Optical Fibre Communications Conference
HCF	Hollow core fibre
TDM	Time domain multiplexing
OXS	Optical cross-point switch
FPGA	Field programmable gated array
NIC	Network interface card
LPFS	Large port-count fibre switch
MCF	Multi-core fibre
DC	Data center
VDC	Virtual data center
ODL	Open Daylight
VLAN	Virtual local area network
DSCP	Differentiated services code point

1.2 Document History

Version	Date	Authors	Comment
ToC v1.0	28/11/2016	See the list of authors	First ToC version
ToC v1.2	21/12/2016		Second ToC version
ToC v2.0	25/01/2017		Final ToC version
Integrated v6	02/06/2017		Integrated version for review
revHCHM	06/06/2017		Reviewed by H.C. Mulvad
revHCHM+MPE	06/06/2017		Reviewed by M.P. Enrico
Final	09/06/2017		Quality check and final version

2 Mid-term scenario: ECOC Demonstrator

The mid-term architecture of COSIGN aims at investigating the benefit of including optical switching into more traditional network architectures. A demonstrator was developed to do both measurements and dissemination of this approach. This demonstrator was developed and successfully showcased at the ECOC conference in Düsseldorf in 2016 and is thus referred to as the *ECOC demonstrator*. The demonstrator was realised according to the plan laid out in [3] and is also described in [4]. Below we will review the configuration of the demonstrator and the ECOC demonstration event and then focus on the outcomes of the characterisations of the use cases.

2.1 Demonstrator configuration

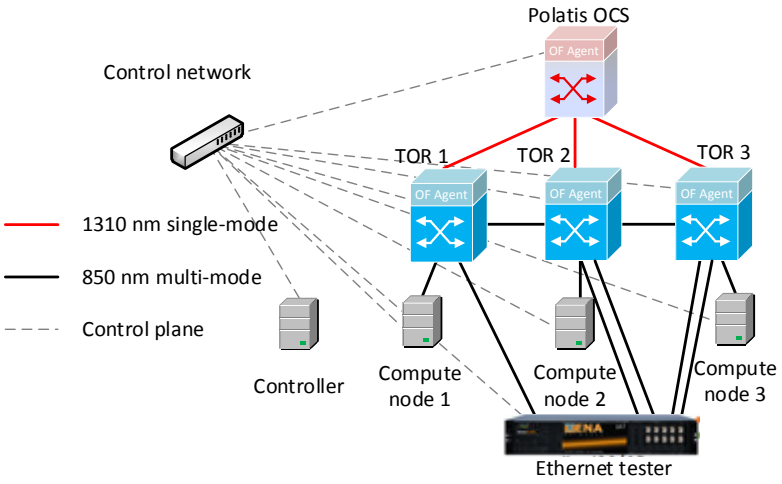


Figure 2-1 Topology diagram of Midterm demonstrator

The used setup shown in Figure 2-1 consisted of 4 server nodes, 3 TOR (Top of the Rack) switches and a single Polatis OCS for optical circuit sharing. One of the server nodes hosted the controller while the remaining three are compute nodes in the network. Each compute node is connected to a separate TOR in order to probe the performance of the network. A Xena Ethernet tester was included in the setup to emulate realistic traffic flows between the three TORs illustrated. All data plane connections are optical 10 Gbit/s interfaces.

A dedicated management network was employed to control and configure all servers and network components in order to implement and execute the different use cases. At the ECOC exhibition both VDC and vApp use cases were demonstrated. Both use cases are discussed in detail in subsequent sections of this document. As this was a demonstration as part of a conference exhibition great emphasis was put on the dissemination of key concepts and main results. Videos were produced of both use cases which explained these concepts and results in an easily accessible way. These videos were looped at the stand while live demonstrations were carried out by project members to offer more in-depth presentation of the implementation details and performance benefits of the use cases.

For showcasing the actual demonstrator at the exhibition all equipment was installed in two 20U rack cabinets, which was fully mounted and tested at DTU and then transported to Düsseldorf. This approach minimized any risk of malfunction of the demonstrator and reduced the setup time at the venue. Figure 2-2 shows the demonstrator equipment as it was presented to the audience. The monitors on the racks showed live demonstrations of one use case and recorded video of the other use case, alternating between which use case was shown live and which was shown on recorded video.

Figure 2-2 – Photos of the demonstrator as it was presented at the ECOC 2016 exhibition.

The demonstrator event was a great success. Both use cases were successfully demonstrated live on alternating schedules and the whole effort generated significant awareness of the efforts and accomplishments in COSIGN. The COSIGN demonstrator occupied half of project partner Polatis' stand at the exhibition, which offered the project great visibility and led to the demonstrator being shown in the ECOC TV webcast.

In the following section we will discuss the two main use cases addressed in COSIGN: the VDC and the vApp and their implementation in the ECOC demonstrator platform. The section will include both the implementation and results which were shown to the audience, but also measurements and characterizations which go beyond what was possible to disseminate at the event. This emphasises that the demonstrator platform has been useful for dissemination of project work but also for investigation and measurement of performance improvements and successful component integration as main project results.

2.2 VDC use case

Besides the pure functional test, performed at the ECOC 2016 demonstrator, which showcased the dynamic on-demand creation and deletion of VDC instances on top of the COSIGN data plane through the collaborative efforts of the orchestration and control planes, we also conducted additional tests to better assess the benefits of the COSIGN approach towards joint resource orchestration.

The additional tests were focused on determining the increased acceptance of VDC instances deployed in a shared physical DC when compared to legacy solutions, where resource provisioning is performed in a two-step fashion without coordination between the VM deployment and the network connectivity configuration. Moreover, we also analysed the scalability regarding the provisioning time of VDC instances when faced with increasing number of demands to serve.

2.2.1 Test plan and results

An emulated environment was used to study aspects of scaling the mid-term architecture to larger size networks. The physical demonstrator platform described above has proven the successful integration of components. An emulated scenario employing the Mininet emulation tool has been constructed to study algorithm behaviour in larger networks. However, apart from the data plane being emulated, all the software layers (orchestrator and controller) are the ones developed within the COSIGN architecture and implemented in the physical testbed. To perform more agile tests, a VDC request generator has been developed, which produces sets of VDC requests according to some input parameters and then sends the VDCs to be instantiated to the algorithms module. The communication between the VDC generator and the algorithms module is performed employing the same interface developed to interconnect the VDC Horizon dashboard and the algorithms module. Figure 2-3 depicts a schematic of this setup.

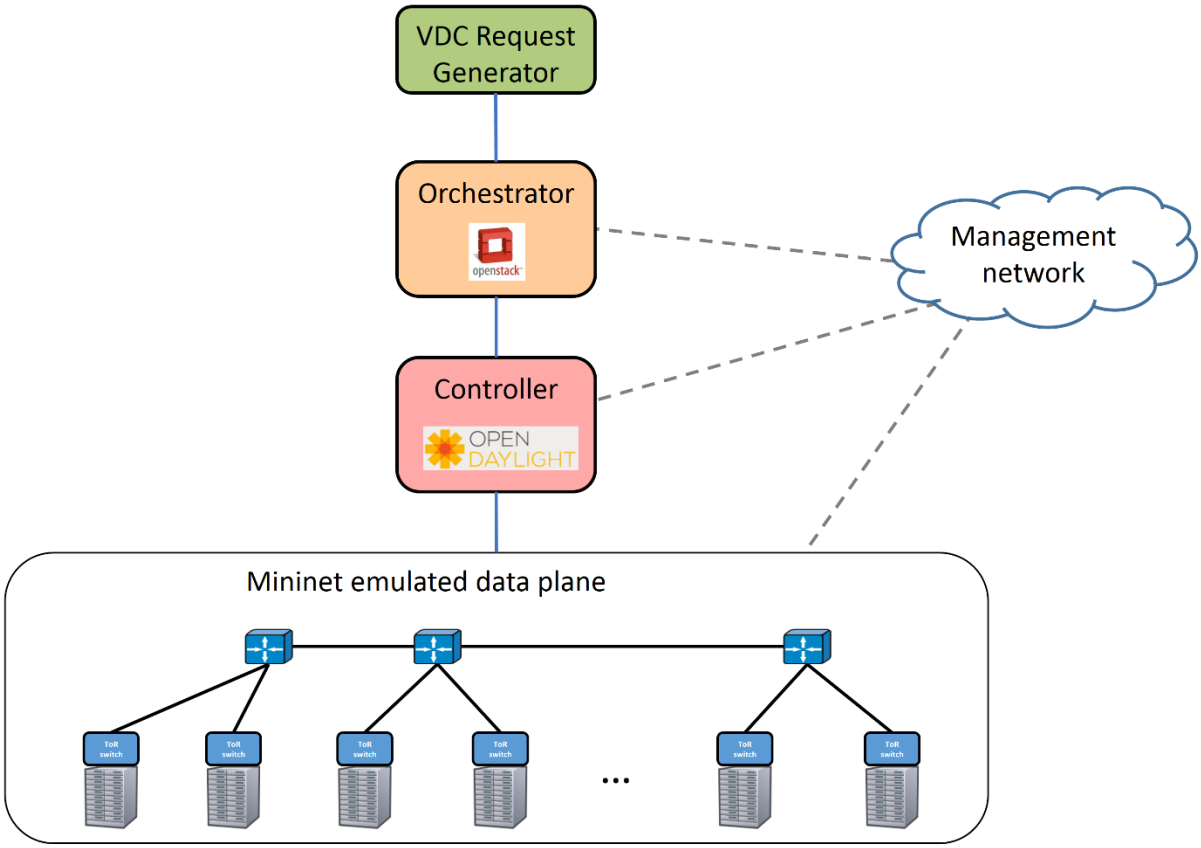


Figure 2-3– Schematic of the employed experimental setup for the VDC use case

Given this setup, we configured a data plane scenario which consisted of a DC composed of three clusters, with each cluster consisting of 7 racks and 40 servers per rack. Then, all the racks are connected to a TUE ToR switch, which are in turn connected to a central Polatis fibre switch, one per cluster. Finally, all Polatis switches are connected following a ring topology. All interconnected pairs of optical nodes at the topology are connected through 24 fibre links. As for the servers, due to hardware limitations, these are also emulated, with each server having 8 CPU cores, 16GB of memory and 1TB of disk available for VM instantiation. In this regard, in our experiments, we will not deploy any VM onto the compute nodes (i.e. the emulated servers). Note, however, that the overall deployment process is effectively the same, with the algorithms module accounting for the emulated servers and their resources when determining the VDC request mapping, that is, into which servers the VMs are deployed and the suitable network path to interconnect them. Figure 2-4 depicts the configuration of the emulated DC scenario.

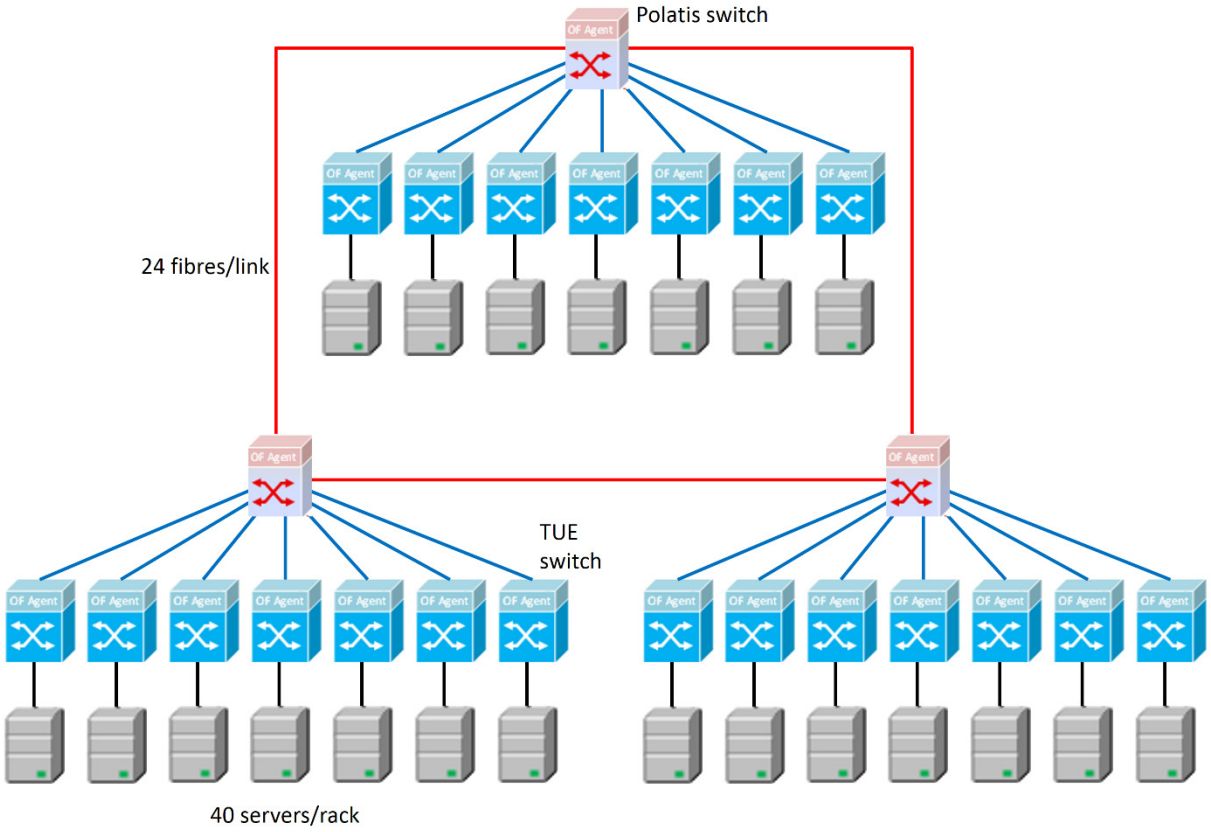


Figure 2-4 – Emulated DC scenario topology.

To analyze the performance of the overall VDC provisioning, we employed the developed VDC generator, which creates random VDC instances which are then sent to the algorithms module. In this regard, we considered a scenario in which a set of VDC requests has to be allocated at the DC physical infrastructure concurrently, with an arbitrary long life time, thus no VDC deletion will be performed (a static scenario is considered). Thus, each of the VDC requests are sent sequentially to the algorithms module to initiate the VDC mapping procedure. Then, the algorithms will calculate both the VM placement and the end-to-end network path between servers to satisfy the virtual links in the VDC. Once the mapping is selected, the network configuration between the end-points corresponding to the emulated servers onto which the VMs would have been deployed is triggered according to the decisions taken at the orchestrator. As said before, to analyze the benefits of a joint resource orchestration, we have also performed the VDC deployment test in a scenario in which the resource mapping decisions are taken separately: on the one hand, the allocation of the VMs is performed solely considering the load of the emulated servers while, on the other hand, the path selection between VMs is performed by the Path Computation Manager (PCM) in the control plane.

For the two scenarios described above (denoted joint and non-joint allocation), in order to extract more meaningful conclusions, different configurations of VDC are generated. In particular, we generated VDC instances consisting of between 2 and 5 VMs interconnected with virtual links in a full mesh fashion. The characteristics of the VMs (i.e. the requested resources) are chosen among the configurations stated inTable 2-1. These VMs are interconnected with virtual links with a requested bandwidth chosen among the set {10, 100, 1000} Mb/s. Given these VDC configurations, we analyzed for increasing sizes of the VDC demand set the number of accepted VDC for both allocation strategies. To achieve a significant statistical relevance, 10 random set repetitions have been averaged per data point in all reported results. Figure 2-5 depicts the evolution of the acceptance of VDC as a function of the demand set.

Table 2-1 – VM configurations for the generated VDCs in the demand set

CPU Cores	Disk (in GB)	Memory (in GB)
1	1	0.5
1	20	2
2	40	4
4	60	8
8	160	16

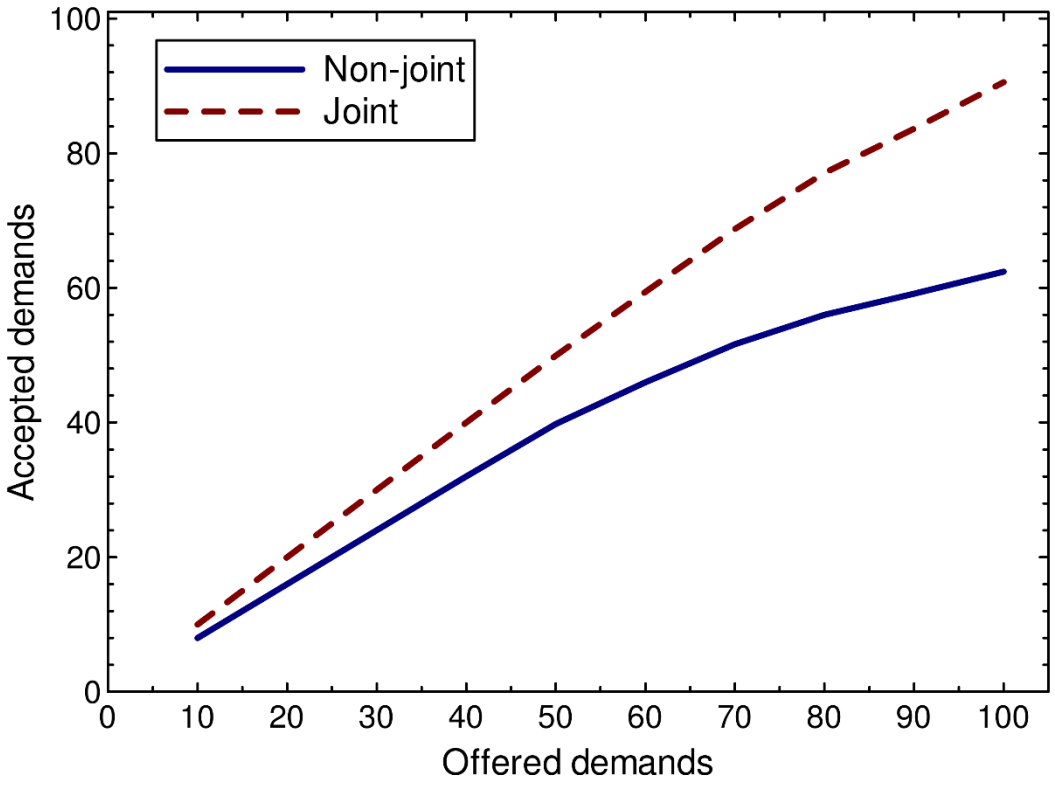


Figure 2-5 – Comparison between non-joint and joint VDC mapping strategies as a function of the offered VDC demands.

It can be appreciated how a joint mapping of the VDC instances allows for a higher number of accepted requests, up to around 42% more when compared with a mapping strategy that does not coordinate both mapping phases. This is due to the network awareness of the node mapping phase in the joint approach, which lowers the chances of blocking due to the lack of network resources during the link mapping phase. To further analyze this phenomenon, we have also extracted results focusing on the characteristics in terms of the number of virtual nodes (i.e. VMs) present in the VDC requests. In this regard, we considered essentially three different scenarios: a) each VDC requests between 2 and 3 VMs; b) each VDC requests between 3 and 4 VMs; and c) each VDC requests between 4 and 5 VMs. In all three scenarios we have particularized the size of the demand set to 100 VDC requests. Such results have been already reported in deliverable D4.5 as the final integration results regarding the orchestrator layer for VDC provisioning [2].

The obtained results confirm the superior performance of the orchestrated joint resource provisioning for all considered configurations. Particularly, we appreciated that in situations with more meshed virtual infrastructures (scenario c), the non-joint approach performs very poorly, since a higher number of virtual links have to be allocated. Because the end-points to be connected (i.e. the servers) are a consequence of a mapping decision considering only the load of the computing resources, the chances that a significant number of end-to-end paths cannot be realized due to not having enough optical network resources increases, affecting the number of successfully allocated VDC instances. On the other hand, in the joint approach this effect is mitigated, thanks to the network awareness of the mapping approach, which selects the placement of the VMs considering the optical resources status. In this way, the acceptance of VDC remains fairly stable across the different VDC configurations.

Next, we analyzed both the differences between the provisioning times between the non-joint and joint approaches to determine the incurred overhead due to the higher complexity of the mapping procedure for the orchestrated resource mapping, as well as the scalability of the algorithms module when facing bigger VDC instances, which require a larger number of calculations to find a suitable mapping of all resources. Table 2-2 shows the necessary time to configure a VDC request in both joint and non-joint approaches. Note that the reported times also include all the necessary delays to guarantee that the datastores in the controller level are properly updated, which is essential for a proper update of the status of the optical resources and a correct configuration of the network connectivity in the data plane.

Table 2-2 – VDC configuration times for both joint and non-joint approaches.

Scenario	Time non-joint (s.)	Time joint (s.)
a	11.99	20.63
b	33.08	48.33
c	49.77	69.43

The obtained times reflect first that the introduced overhead due to the joint resource provisioning (mainly due to the algorithms module) is fairly non-relevant (a maximum difference of about 20 seconds is observed) taking into account the increased VDC acceptance that can be achieved when compared to a legacy non-joint provisioning approach (up to two times in the worst-case scenario). Additionally, it can be appreciated that increasing the size of the VDC requests has an almost linear effect on the time required to calculate the VDC mapping, highlighting the scalability of the algorithms module.

Finally, to conclude our studies, to further evaluate the benefits of an orchestrated approach to VDC mapping, we also performed some tests considering a dynamic scenario, where VDC requests arrive at the DC following a random arrival and departure process. For this, we consider a Poisson arrival process, with exponentially distributed inter-arrival times (IATs) and holding times (HTs). We evaluated the blocking probability of the VDC requests considering increasing loads. For this, we have fixed the average IAT to one time unit and increased the average value of HT. Moreover, we considered the initial scenario in regards of VDC configuration, that is, with VDC requesting between 2 and 5 VMs each. *Figure 2-6* depicts the obtained results. All the data points have been extracted considering $2 \cdot 10^5$ random VDC arrivals. The obtained results further confirm the benefits of a joint mapping, achieving blocking figures with up to 50% reductions when compared to the non-joint approach.

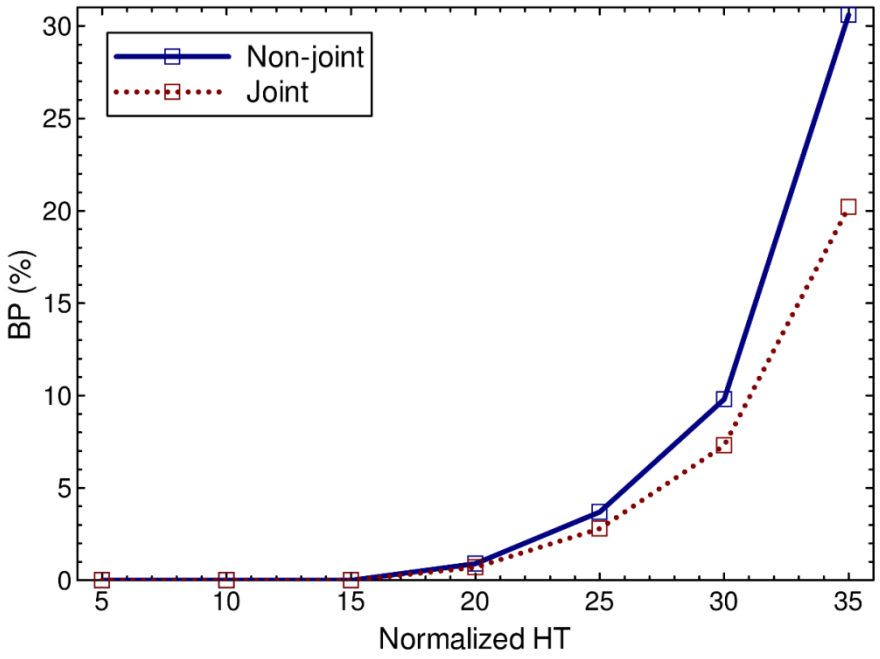


Figure 2-6 – Comparison of blocking probability (BP) between no joint and joint VDC mapping strategies as a function of the normalized HT.

2.3 vApp use case

The vApp use case utilises the introduction of an SDN controlled optical circuit switch into the network to adaptively establish connections to alleviate congested connections in the static part of the network. The use case has been demonstrated on the mid-term demonstrator platform presented at ECOC. Subsequently an implementation with upgraded interface speeds was investigated to provide further data on the prospects of this approach.

2.3.1 Test and results from the ECOC demonstrator implementation

Figure 2-7 depicts the hardware (HW) setup for vApp demo, which includes: 3 servers (IBM x3690 X5), 3 TU/e ToR switches, one Polatis switch and one Xena Ethernet traffic generator/tester – all connected by 10G optical links; and a controller with a separate control network – as described at the beginning of this section.

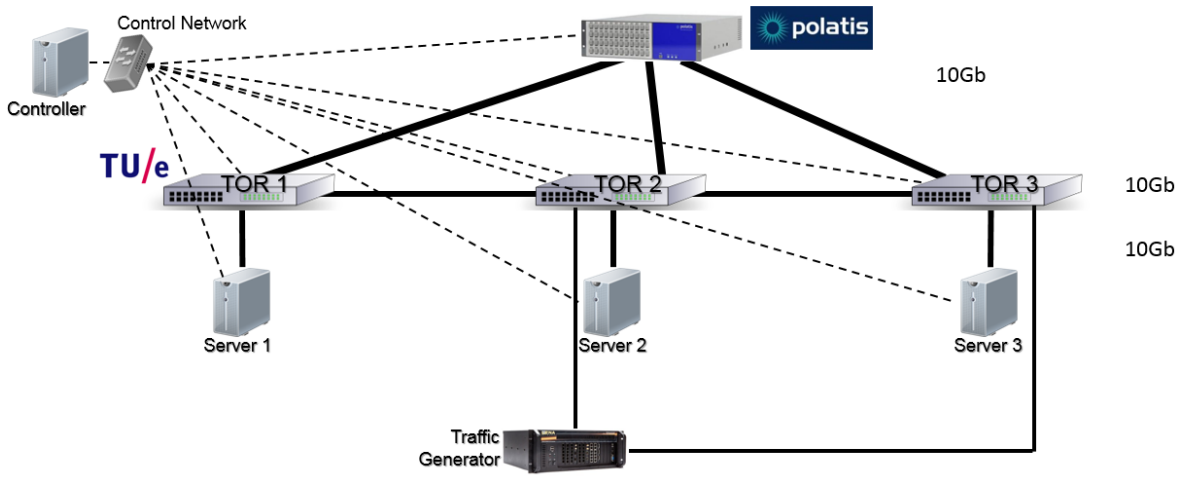


Figure 2-7 - ECOC demo HW setup showing the configuration of servers, switches and the traffic generator

The vApp demo demonstrates the benefits of the shared optical circuit as compared to a private circuit. The demonstrated scenario includes:

- Two elephant flows: one between servers 1 and 3, and another between servers 2 and 3.
- Optical circuit between TOR 2 and TOR 3.
- Mice flows between TOR 2 and TOR 3 through the direct packet plane link, by the traffic generator.

The elephant flows are sampled by sFlow, detected, and DSCP tagged by the virtual observer employed on each server.

Two different scenarios were investigated for the vApp use case: private circuits and shared circuits.

Private Circuit:

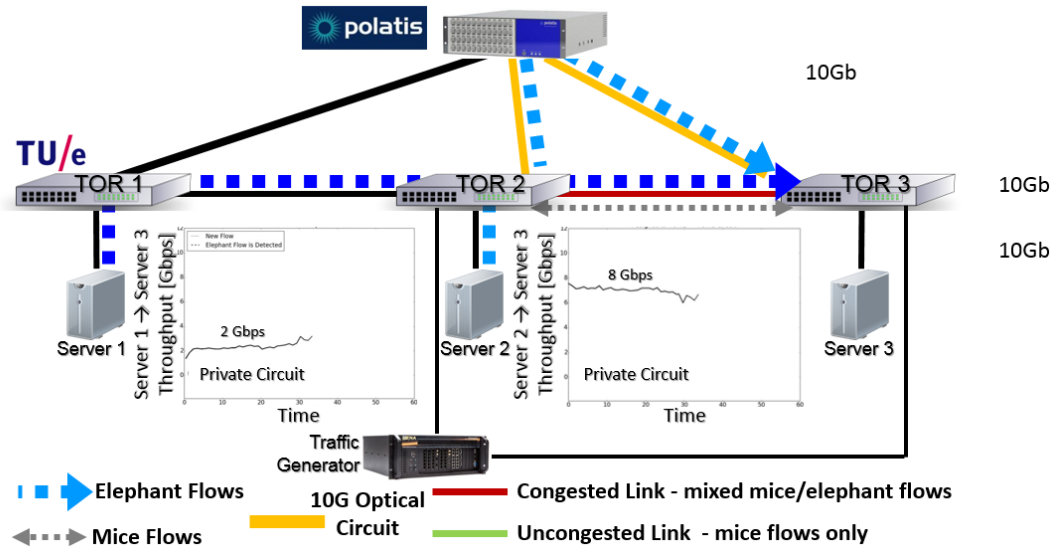


Figure 2-8 – Private circuit scenario

Figure 2-8 presents the private optical circuit scenario, in which the elephant flow between servers 2 and 3 is transmitted through the established optical circuit with high throughput (~8Gbps). Whereas the elephant flows between servers 1 and 3 are transmitted through the packet plane, along with the mice flows over the congested link between TOR 2 and TOR 3; therefore, its throughput is much lower (~2 Gbps).

Shared Circuit:

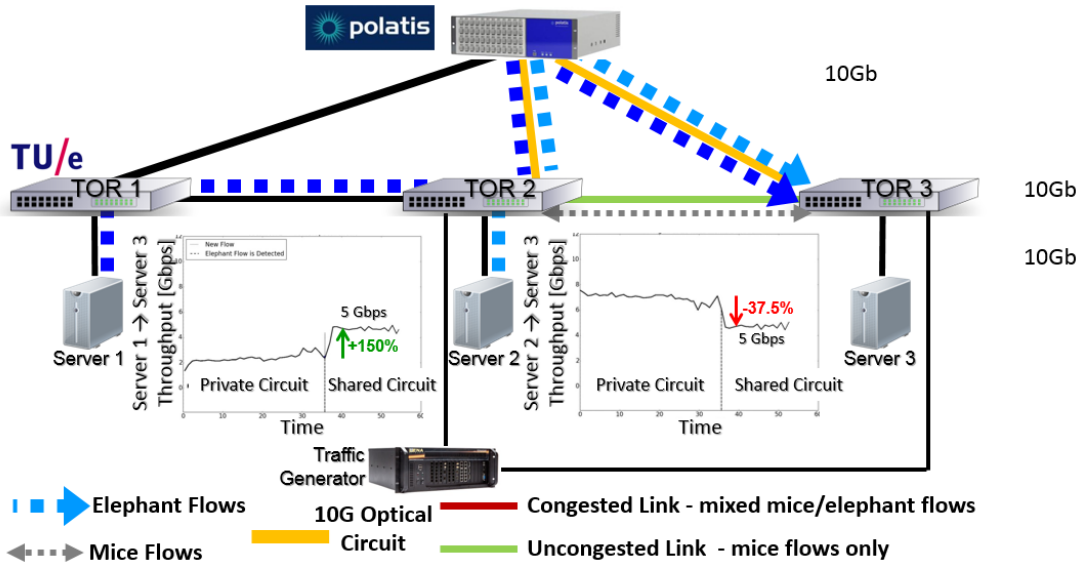


Figure 2-9 – Shared circuit scenario

Figure 2-9 presents the shared optical circuit scenario. Both elephant flows are transmitted through the established optical circuit. Therefore, the elephant flows (transferring 1GB by TCP) fairly share the optical circuit bandwidth. In this scenario, the throughput of the elephant flow between server 1 and 3 is increased by 150%, whereas the throughput of the elephant flow between server 2 and 3 is decreased by 37.5%. Furthermore, the optical circuit is fully utilized, and improved by 20% as compared to the private circuit case.

2.3.2 Improved Demo with 40Gbps Optical Plane

In the ECOC demo, we showed how the shared optical circuit can offer fairness in terms of elephant flow throughputs. However, due to the limited 10Gb/s optical circuit we were unable to demonstrate the full potential of vApp. In the improved demo at TU/e we use the same topology but with 40Gb optical links between the TORs and the Polatis switch – benefitting from the optical transparency of the fibre switch making it agnostic to the rate of the switched connections. Furthermore, we also measured the RTT of the mice flows over the packet network plane.

Throughput Comparison:

Figure 2-11 presents the elephant flow throughputs for both private and shared optical circuits (which are alternately configured). The throughput of the elephant flow between server 2 and 3 is not affected by the optical circuit type. On the other hand, the throughput of the elephant flow between server 1 and 3 is greatly improved by the shared optical circuit (by 400%), and reach 10Gbps throughput which is the maximum available bandwidth of the servers' NIC. This improvement is accomplished due to higher link bandwidth of the optical plane.

Private Circuit:

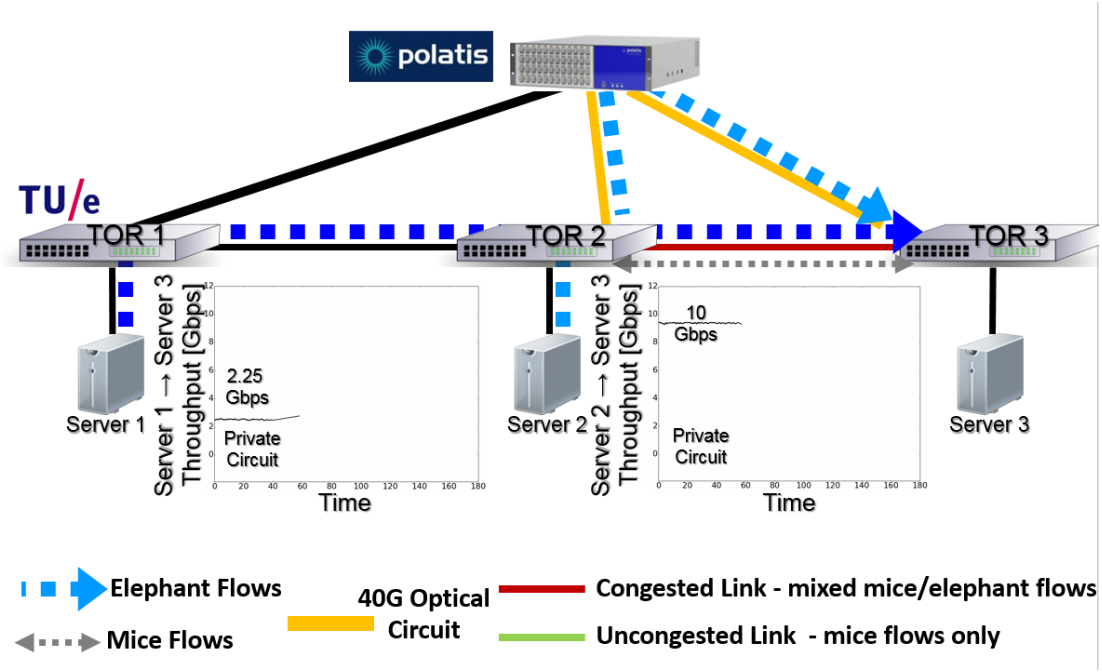


Figure 2-10 – Throughput obtained by private circuit configuration

Shared Circuit:

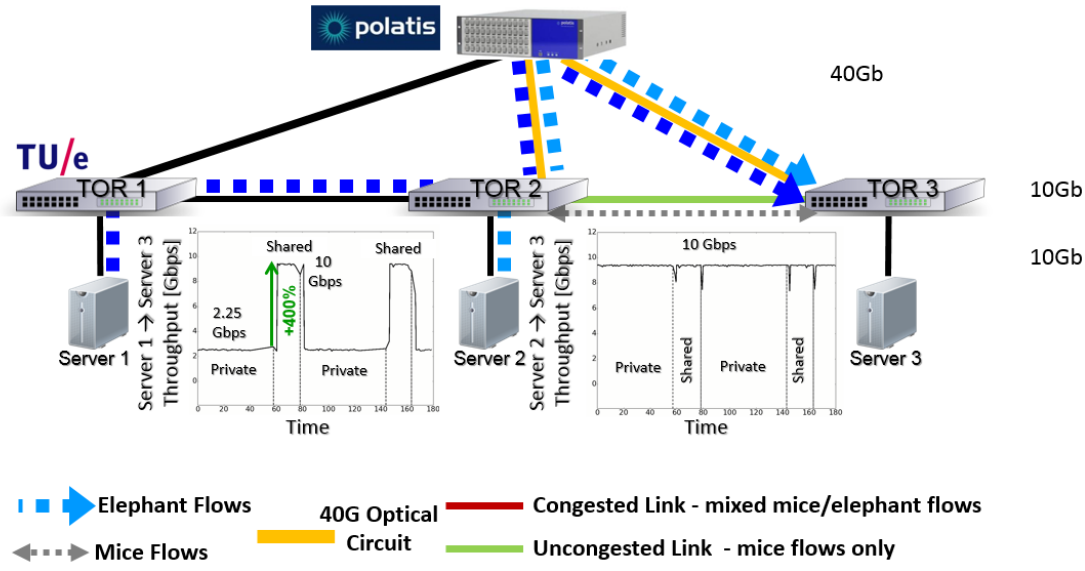


Figure 2-11 – Throughput comparison between private and shared optical circuits

RTT Comparison:

Figure 2-13 presents the RTTs of the mice flows through the packet network plane, for both private and shared optical circuits (which are again alternately configured). The RTT of both mice flows is decreased by approximately 50% when using the shared circuit. Furthermore, the RTT of the mice flows between server 2 and 3 fluctuates more, since it interacts with two TCP sessions (elephant flow between server 1 and 3, and between server 2 and 3). On the other hand, the mice flow between server 1 and 3 interacts only with a single TCP session.

Private Circuit:

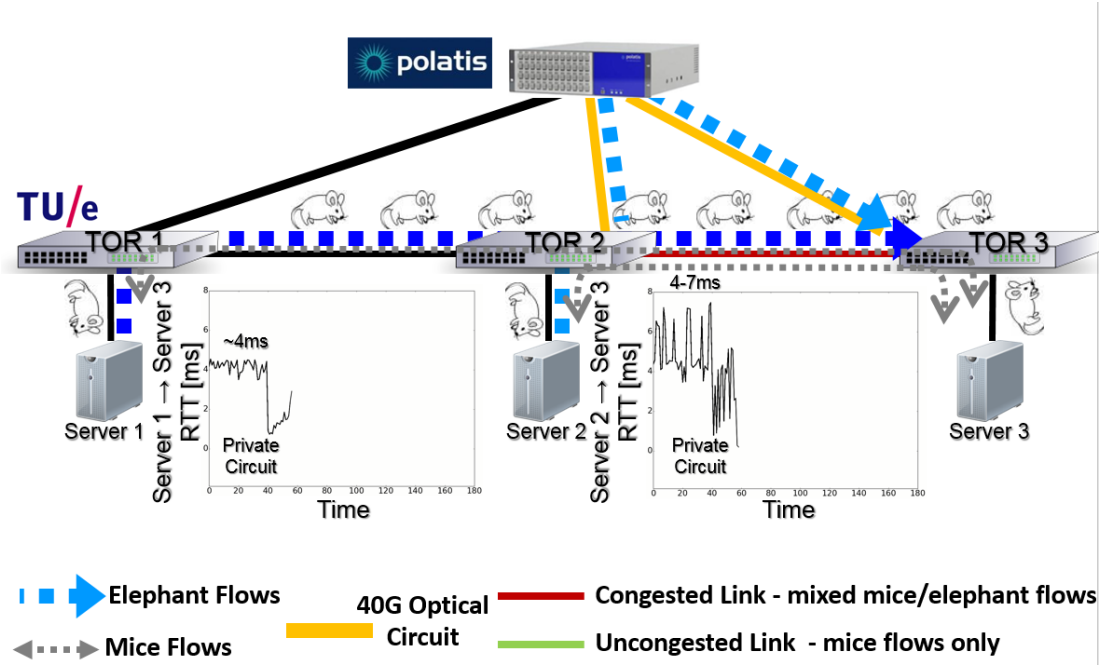


Figure 2-12 - RTT obtained by private optical circuit configuration

Shared Circuit:

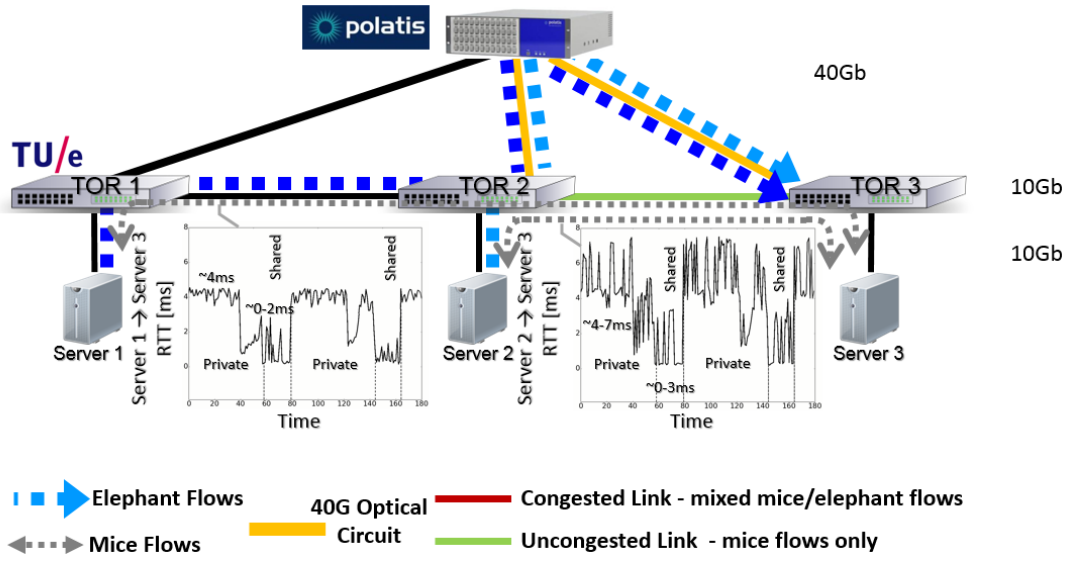


Figure 2-13 - RTT comparison between private and shared optical circuits

2.3.3 Elephant Flow Detection Time

The elephant flows are detected by the virtual observer, which is employed on every server. Initial flows are assumed to be mice flows, until the transmitted bytes of the flow exceed a given threshold. Then, the flow is tagged as elephant flow, using a pre-defined DSCP value. Therefore, the time to detect an elephant flows mostly depends on the allocated throughput of the flow, which determines the time until the transmitted bytes threshold is reached.

In general, the time to detect elephant flows equals:

$$t = \frac{elephant{-}threshold\ [bits]}{elephant{-}throughput\ (over\ EPS)[bits/sec]}$$

In our demo, the elephant flow threshold is set to 0.5MB, and the allocated throughput of the elephant flows when assigned to the electrical packet plane is approximately 2 Gbps. Therefore, it takes around 2ms to detect an elephant flow.

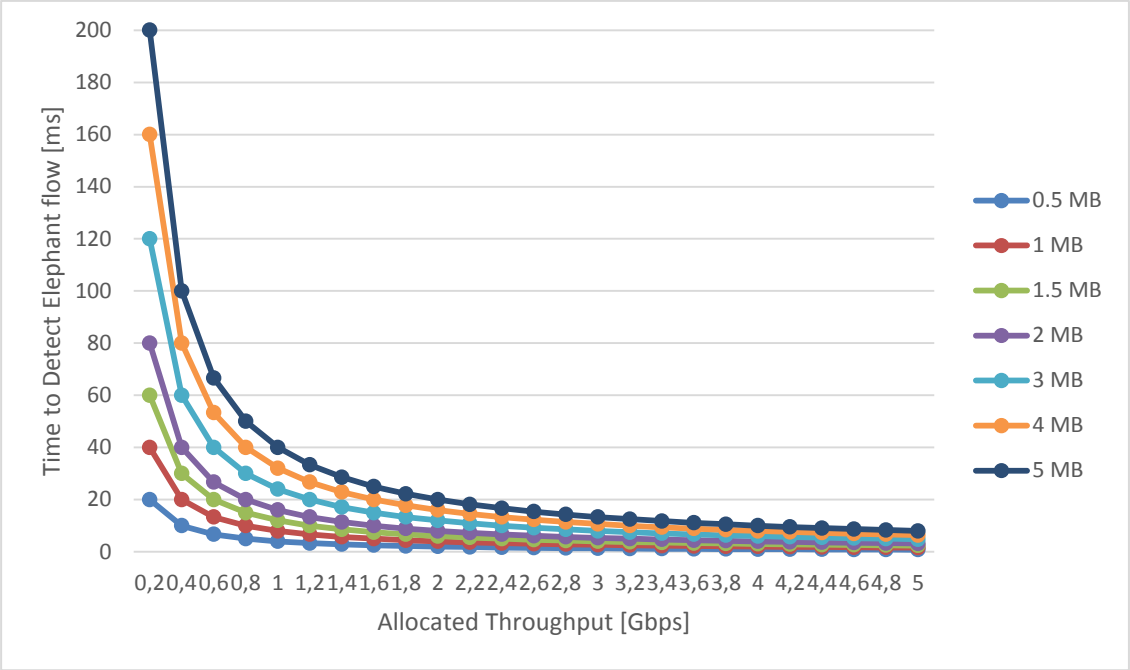


Figure 2-14. Elephant flow detection time vs. allocated throughput and elephant flow threshold.

Figure 2-14 presents a graph of the time it takes to detect an elephant flow vs the allocated throughput of the elephant flow over the electrical packet plane for various values of the elephant flow thresholds. As can be seen, the detection time varies from 200 ms to a few ms. In general, the allocated throughput is assumed to be bigger than 1 Gbps, and the typical elephant flow threshold is 1 MB. Therefore, the detection time should take 8 ms at most.

2.3.4 Conclusions on the vApp use case

The characterization and profiling of network traffic plays an essential role for an efficient management of a generic data network. In the context of an intra-DC environment the majority of TCP flows tend to be short because they are mostly related to bursty network traffic or to latency-sensitive applications, whereas the majority of packets are related to a few large flows (namely elephant flows).

The characteristics of these kinds of flows can be summarized in three broad categories based on their duration and data rate:

- Short duration flows with high data rate
- Long duration flows with low data rate
- Long duration flows with high data rate.

The elephant flows are generally related to large data transfers such as backups or data migrations that can be generated both by the DC's customers and by the DC's owner for different kind of back-end operations. The creation of these flows has an obvious impact on the overall DC network performance, since they tend to congest end-to-end network buffers thus introducing significant delay to all the latency-sensitive mice flows sharing the same buffer, leading to performance degradation of the network.

The efficient management of elephant flows in a legacy DC network is affected by the lack of a centralized, high-level point of view on the whole network infrastructure. Even in a SDN-compliant DC network the physical switches are equipped with powerful dedicated ASICs for data plane switching, but weak CPUs for control plane or any other SDN defined task. In this context, the features provided by the COSIGN orchestrator for the vApp scenario bring a remarkable contribution to the optimization of elephant flow management. The optimization process starts from the interaction between the data plane through the physical and virtual observer that guarantees a continuous monitoring of the topology and of the flows crossing the physical and virtual layer network. The information retrieved by these two orchestrator components is provided to the Orchestrator Algorithms module which is in charge of configuring the optical DCN's circuits. This centralized approach together with the high configurability of the COSIGN orchestrator, allows the DC owner and operator to fully exploit the network capabilities, increasing its usability and overall performance.

Being more practical, an effective management of elephant flows has an immediate and positive impact from the operational, and thus business, point of view. For example, many different data intensive applications e.g. database synchronization, backups, data transfer for back-end operations, data migration or VM migrations can be delivered, thus minimizing the impact on all the other traffic flows and preventing the possibility of SLA breach. vApp requires a specific data-plane topology, and minor changes in the control plane. In turn, it offers substantial improvement in the network performance, both in terms of throughput and RTT.

3 Mid-term scenario: VDC Industrial validation

3.1 Scenario

The original plan reported in D5.3 [3] was to host the COSIGN testbed in Interoute's PoP in Milano Caldera, but due to internal company reasons arising during the last year, the co-location area reserved for research and development activities was moved to the PoP in Pisa with the same conditions, equipment and infrastructure described in [3] and D5.0 [15]. The testbed infrastructure deployed is made of two different macro components: i) the Interoute testbed made of two servers and one Cisco Catalyst switch (described in [15]), and ii) the COSIGN dataplane made of three Tu/E switches, one Polatis switch and one optical converter (Fiber24 Converter). The physical servers host Virtual Machines (VMs) running the COSIGN cloud orchestrator and control plane components (OpenDaylight and OpenStack Controller Node) and the OpenStack computing nodes, which emulate the datacentre servers.

The high-level design is reported in Figure 3-1, and a photo of the physical testbed deployment is shown in Figure 3-2.

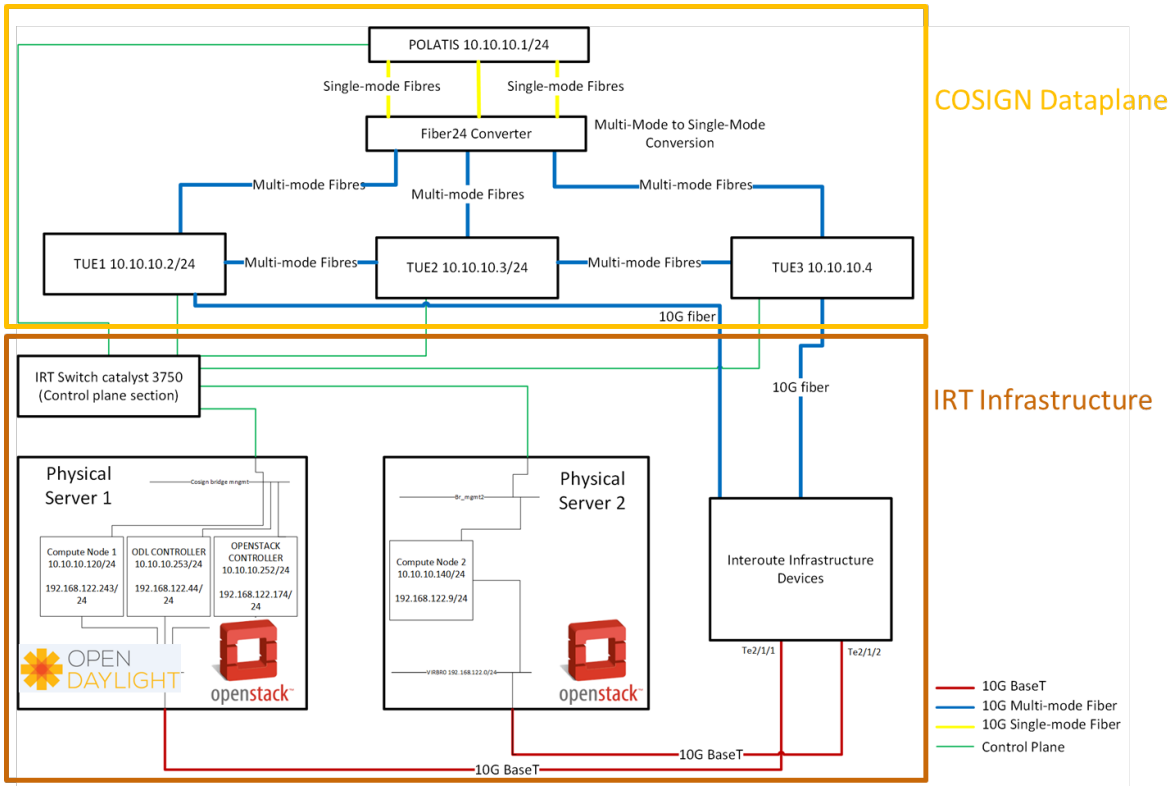


Figure 3-1: Industrial demo, high-level design

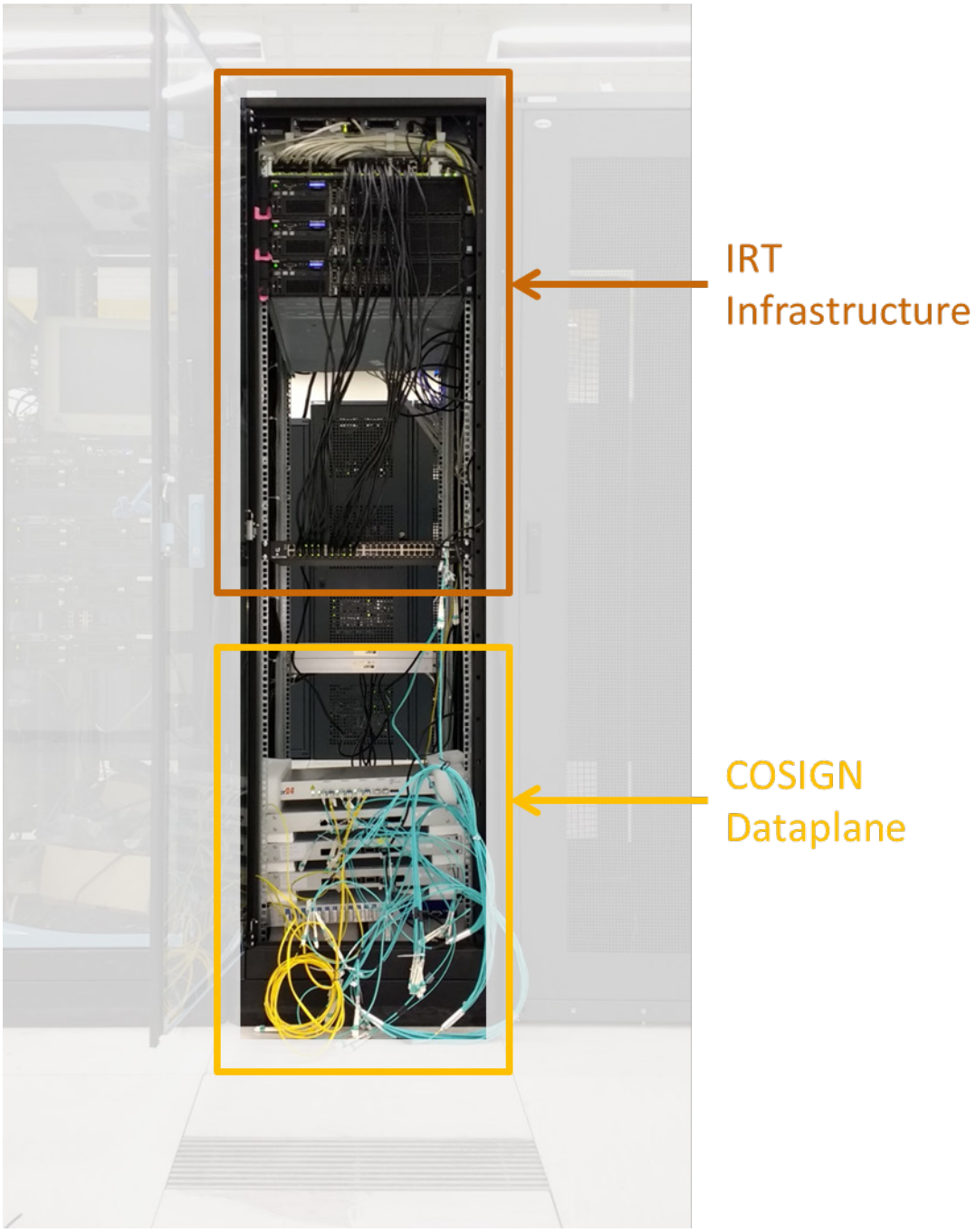


Figure 3-2: Industrial demo, physical deployment

A more detailed description of the physical deployment is provided in the following list:

- Physical Server 1: This server contains three different VMs:
  - OpenStack Controller: this VM is the controller of the COSIGN OpenStack environment, which includes the VDC algorithms module and the COSIGN Horizon dashboard (developed in WP4).
  - ODL Controller: this VM runs the COSIGN OpenDaylight controller, with the COSIGN extensions and modules implemented in WP3 in support of VDC use case in medium-term scenarios (i.e. in a COSIGN DCN composed of TUE and Polatis devices).
  - Compute Node1: This is one of the compute nodes that compose the OpenStack IT environment and represent a DC server.

- Physical Server 2: this server contains one VM:
    - Compute Node2: This is one of the compute nodes that compose the OpenStack IT environment and represent a DC server.

The internal connections between the different components listed above have been made using two Linux virtual bridges in the physical servers. The *management bridge* provides the control plane connectivity between the ODL controller, the OpenStack controller, the two compute nodes and the Catalyst switch. The *dataplane bridge* provides connectivity between all the VMs and the COSIGN data plane (see Figure 3-1).

3.2 Test plan and results

The testing process has been organized in three different phases in order to verify functionalities and performance of all the involved technologies and features. The testing methodologies and all the measurements have been performed following the best practice procedures agreed together with Interoute's internal technical department.

The above mentioned three phases are as follows:

- Data plane tests, described in section 3.2.1.
- COSIGN VDC service tests, described in section 3.2.2.
- Control plane scalability tests, described in section 3.2.3.

3.2.1 Data Plane test

In the Data Plane tests phase we have focused our test activity on the functionalities and performance of the COSIGN network data plane's equipment and on the physical layer of the COSIGN framework, without considering the deployment of the COSIGN orchestrator and control plane components or VDC instances running in the virtualized infrastructure. This means that the tests operate directly at the physical infrastructure and the results are not influenced by the virtualization environments deployed in the two physical servers.

The COSIGN data plane equipment have been delivered to the Pisa Interoute office and installed in our testing lab. The components under test are the following:

- One Polatis switch, acting as a POD switch in the COSIGN test deployment.
- One Optical converter model Fiber24, which provides the multi-mode to single-mode conversion between the Polatis switch and the TUE switches.
- Three TUE switches acting as TOR switches in the COSIGN test deployment.

The tests performed have been organized in two different scenarios based on the data-paths configured on the different data plane devices. In Scenario 1 we considered the communication through the three TUE switches (via TUE switches), while in Scenario 2 we considered the communication through two TUE switches and the Polatis switch (via the Polatis switch).

The data plane topology, with the IDs of the ports of the various components, is represented in Figure 3-3.

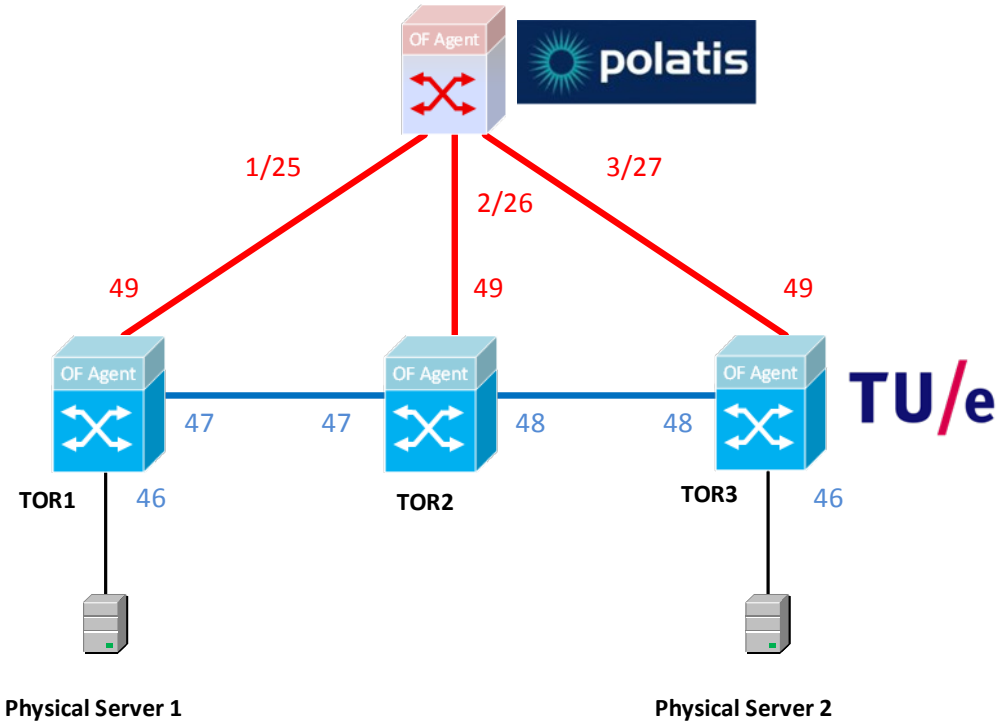


Figure 3-3: Data plane test architecture

The test procedures have been organized as follows:

1. Configuration of the flows across the equipment.
2. Round Trip Time (RTT) test: using ICMP messages from Server 1 to Server 2 and vice versa.
3. Bandwidth test: using Iperf3 tool sending traffic from Server 1 to Server 2 and vice versa.

The measurements have been performed for Scenario 1 and 2 as described below.

3.2.1.1.1 Flow configuration

Scenario 1 consists of a network connection across 3 TUE switches. The TUE switches have been pre-configured, using the Provisioning Manager module in the COSIGN SDN controller, to open a path from server 1, to TUE 1, forwarding to TUE 2, then to TUE 3 and then to server 2; the same for the reverse path.

Scenario 2 consists of a TUE-Polatis connection. Server 1 is still connected to TUE 1, but in this case TUE 1 forwards traffic to the Polatis, then to TUE 3 (skipping TUE2) and then to Server 2. As before, also the reverse path has been configured. The cross connections configured in the Polatis switch, as resulting from the Polatis management interface, are shown in Figure 3-4.

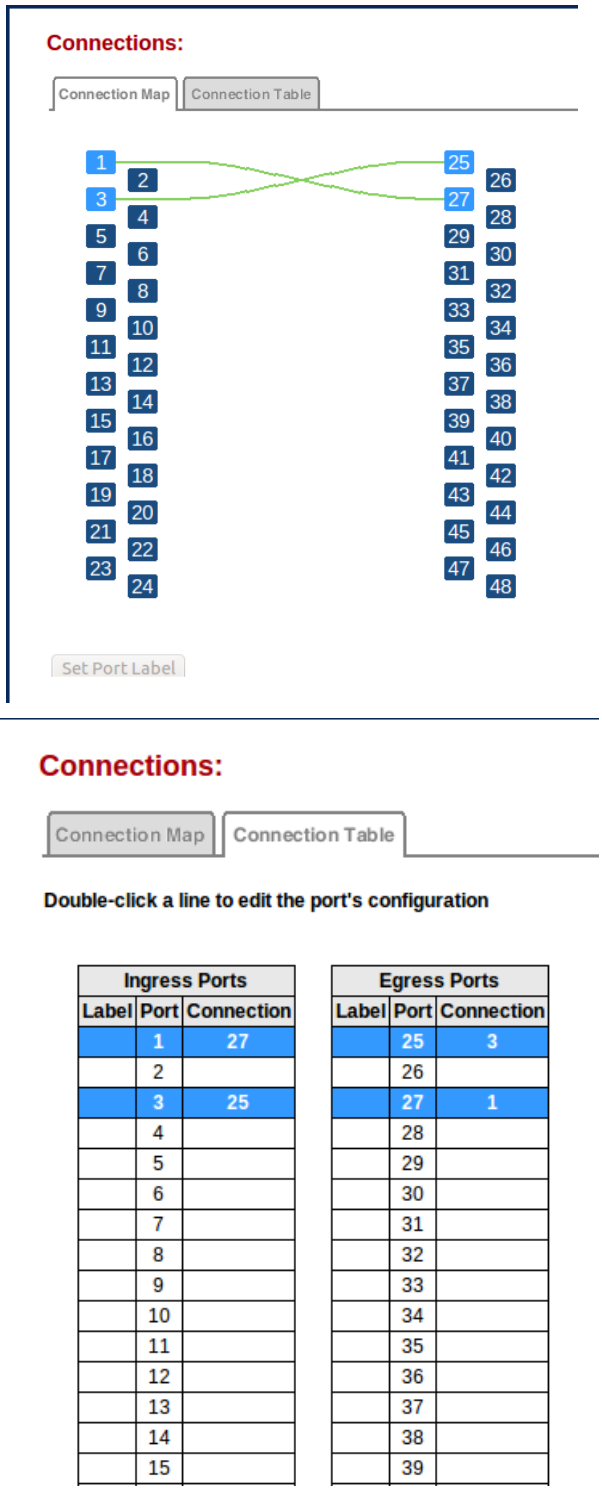


Figure 3-4: Screenshot of the Polatis GUI reporting the cross-connection's configuration

3.2.1.1.2 Round Trip Time latency test

The RTT test has been performed using ICMP messages between the 2 servers. This test has been repeated for 10 times with a ping count of 10, for both scenarios and for both traffic directions. The data obtained have been elaborated and aggregated to show the minimum, maximum and average values as reported in sec. 3.2.1.2.

3.2.1.1.3 Throughput test

The throughput test has been performed using the *Iperf3* tool, following these steps:

1. Set Server 2 as *Iperf3* server running `Iperf3 –s –logfile Iperftest` in order to receive the data sent by Server 1 and write the output to a log file.
2. Set Server 1 as *Iperf3* client running `Iperf3 -c –t 60 –B 172.16.1.2` in order to transmit TCP traffic for 60 sec to the Iperf3 server using the 10Gbit interface with address 172.16.1.2.

The test has been repeated for 10 times in both directions and for both scenarios. The raw data collected have been elaborated and aggregated in order to show the minimum, maximum and average values of throughput. These results are presented in sec 3.2.1.2.

3.2.1.2 Results

In this section, we present the charts summarizing the aggregated results of the experiments carried out in the Interoute testbed and described in the previous sub-sections.

3.2.1.2.1 RTT tests results

Figure 3-5 and Figure 3-6 show the average, maximum and minimum values of the RTT measured in the 10 series of scenario 1 tests (connection across TUE switches), while Figure 3-7 and Figure 3-8 show the same values for scenario 2 (connection across TUE switches and Polatis switch). The average is in the order of 0.2 ms, while the maximum values are in the order of 0.25 ms, without significant differences between scenario 1 and scenario 2.

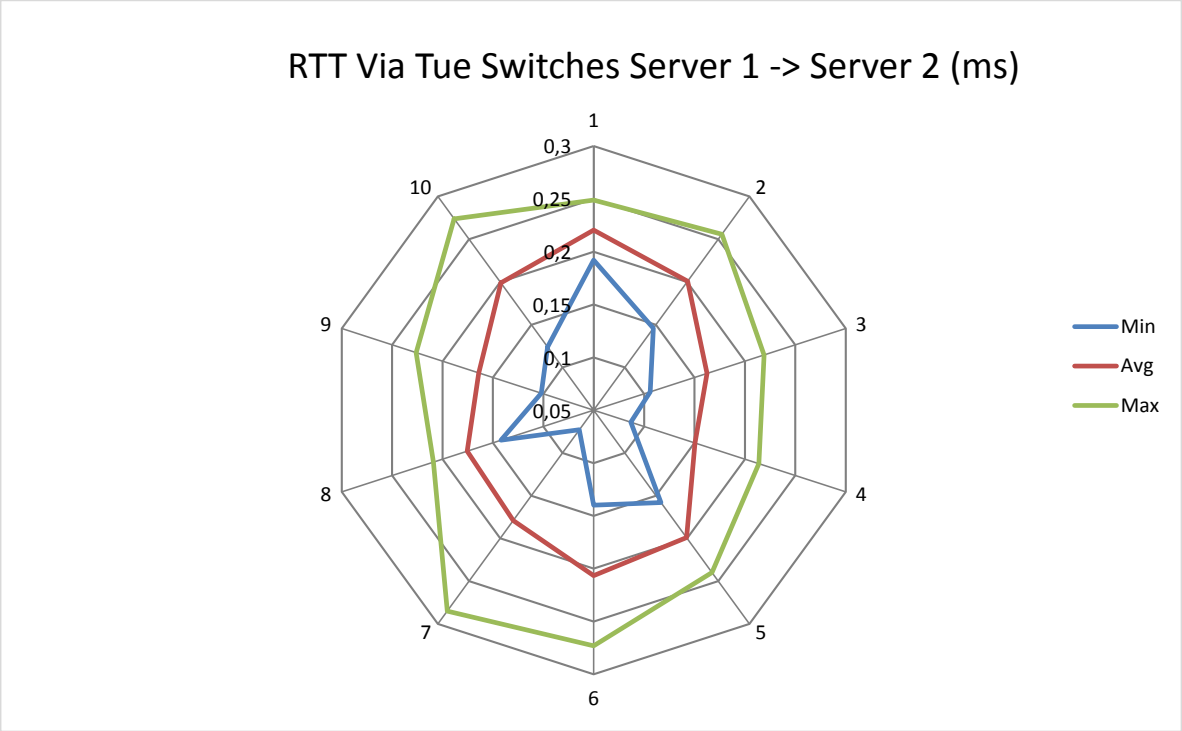


Figure 3-5: Round trip time via TUE switches Server 1 -> Server 2

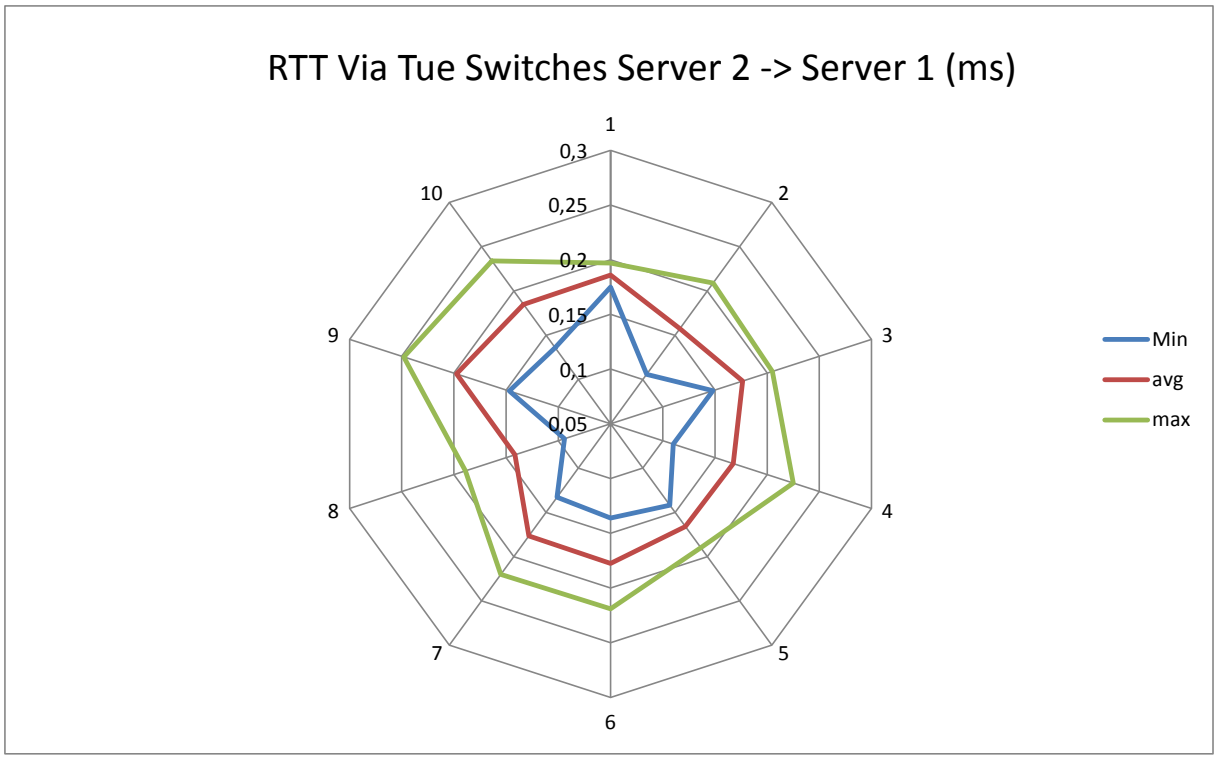


Figure 3-6: Round trip time via TUE switches Server 2 -> Server 1

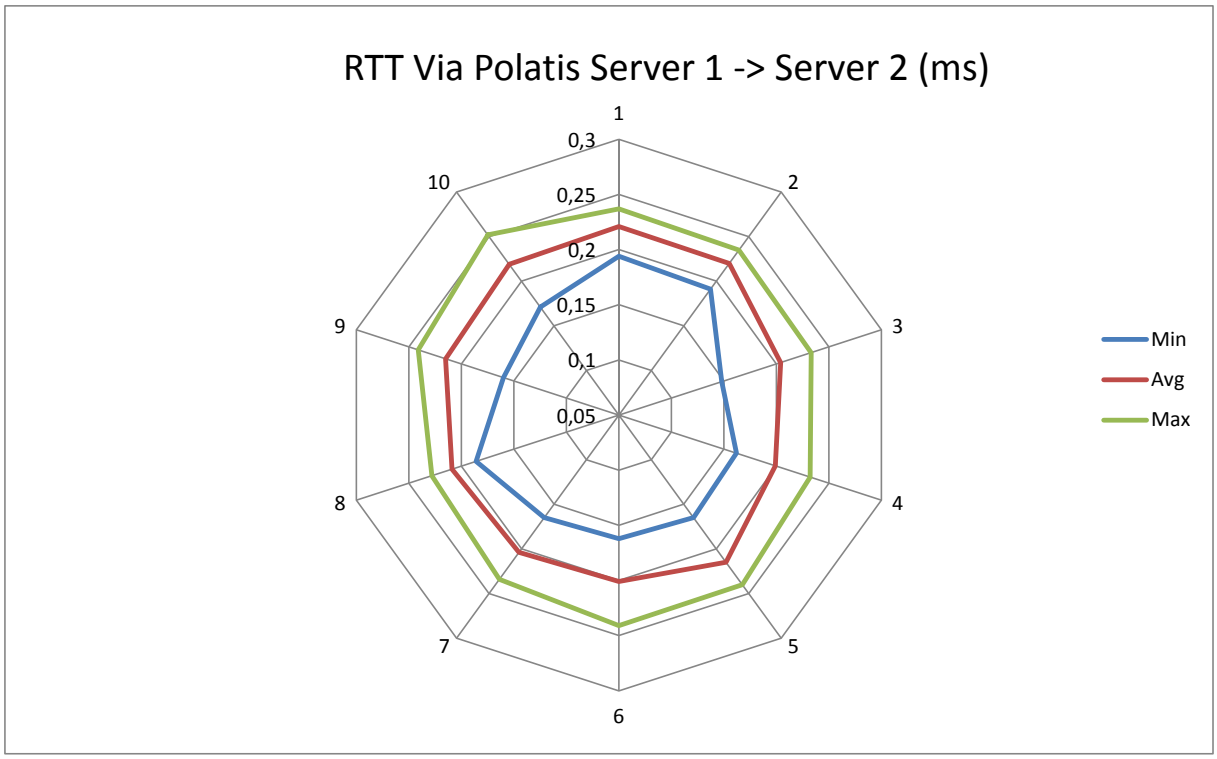


Figure 3-7: Round trip time via Polatis Server 1 -> Server 2

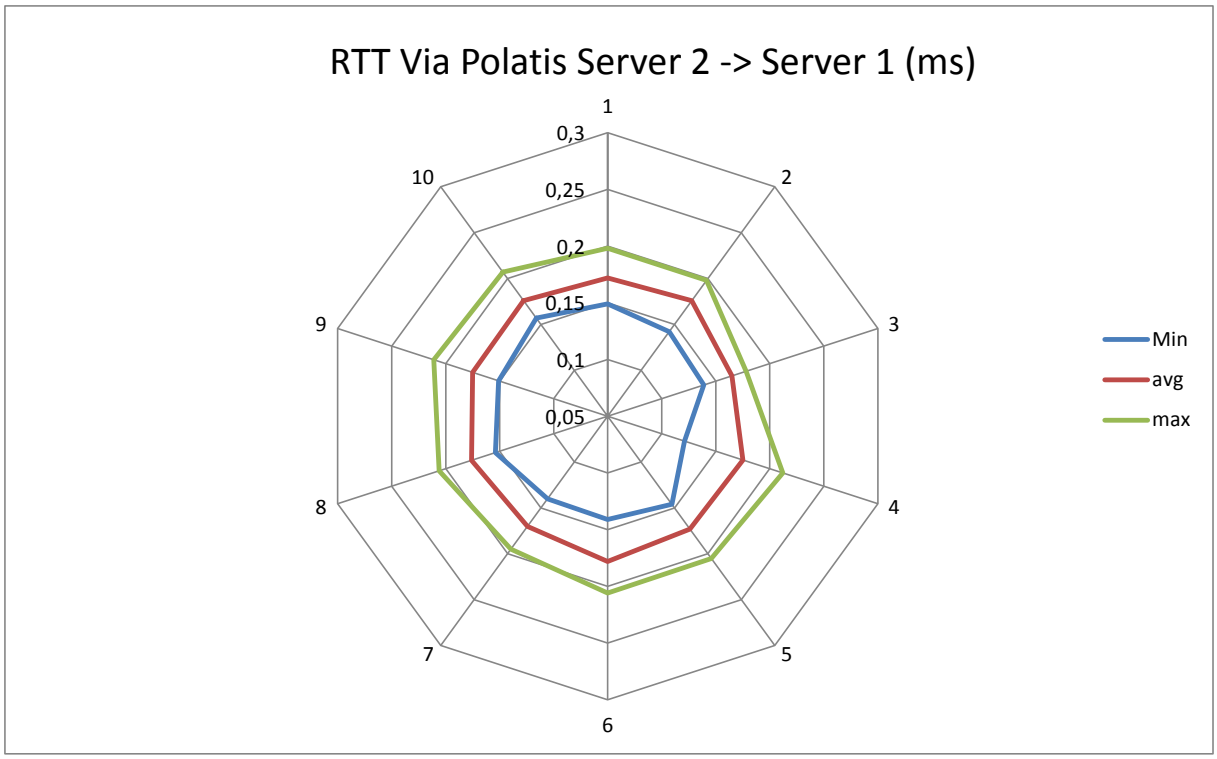


Figure 3-8: Round trip time via Polatis Server 2 -> Server 1

3.2.1.2.2 Throughput tests results

Figure 3-9 and Figure 3-10 show the average, maximum and minimum values of the throughput measured in the 10 series of scenario 1 tests (connection across TUE switches), while Figure 3-11 and Figure 3-12Figure 3-8 show the same values for scenario 2 (connection across TUE switches and Polatis switch). In both cases, the average is in the order of 9.6 Gbit/s, while the maximum values are in the order of 9.8 Gbit/s.

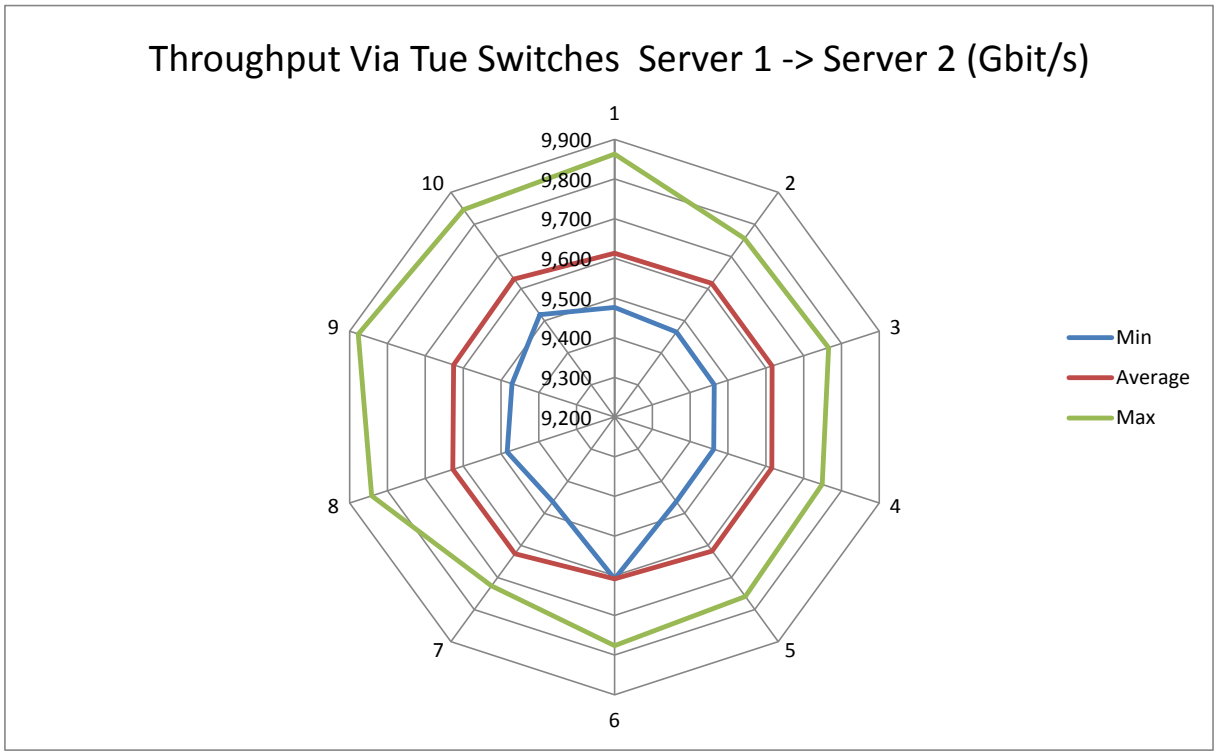


Figure 3-9: Throughput via TUE switches Server 1 -> Server 2

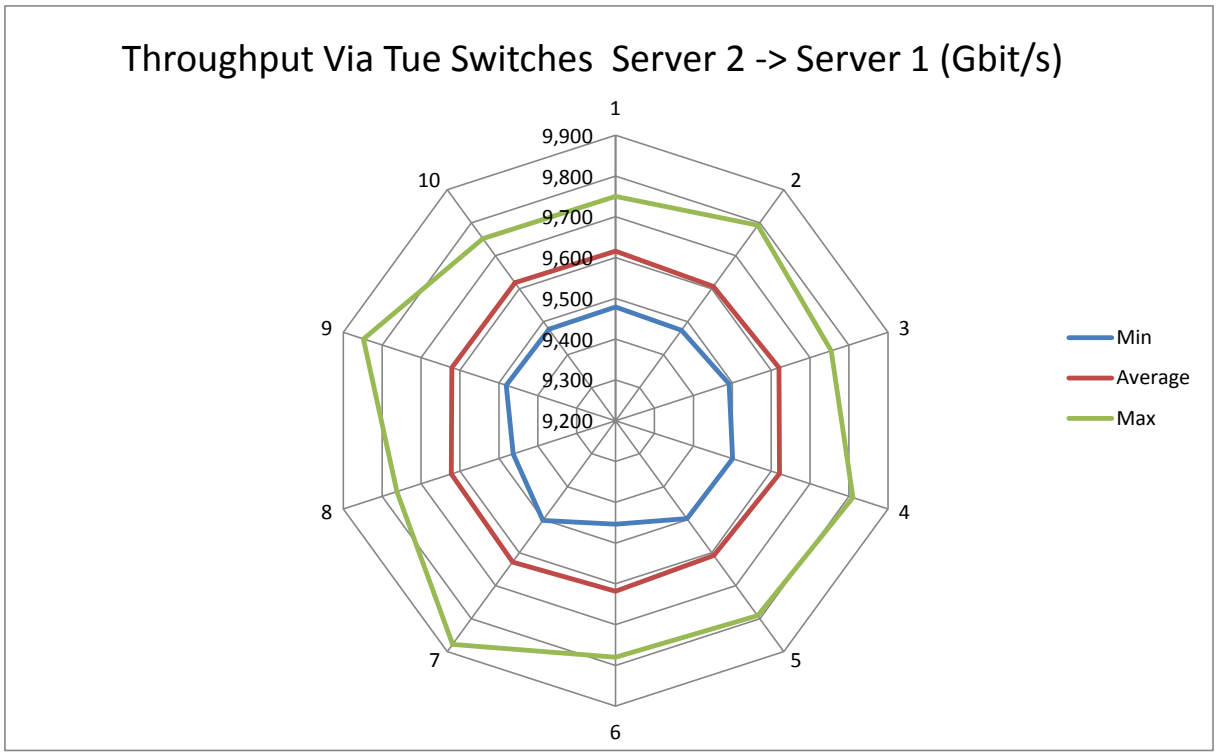


Figure 3-10: Throughput via TUE switches Server 2 -> Server 1

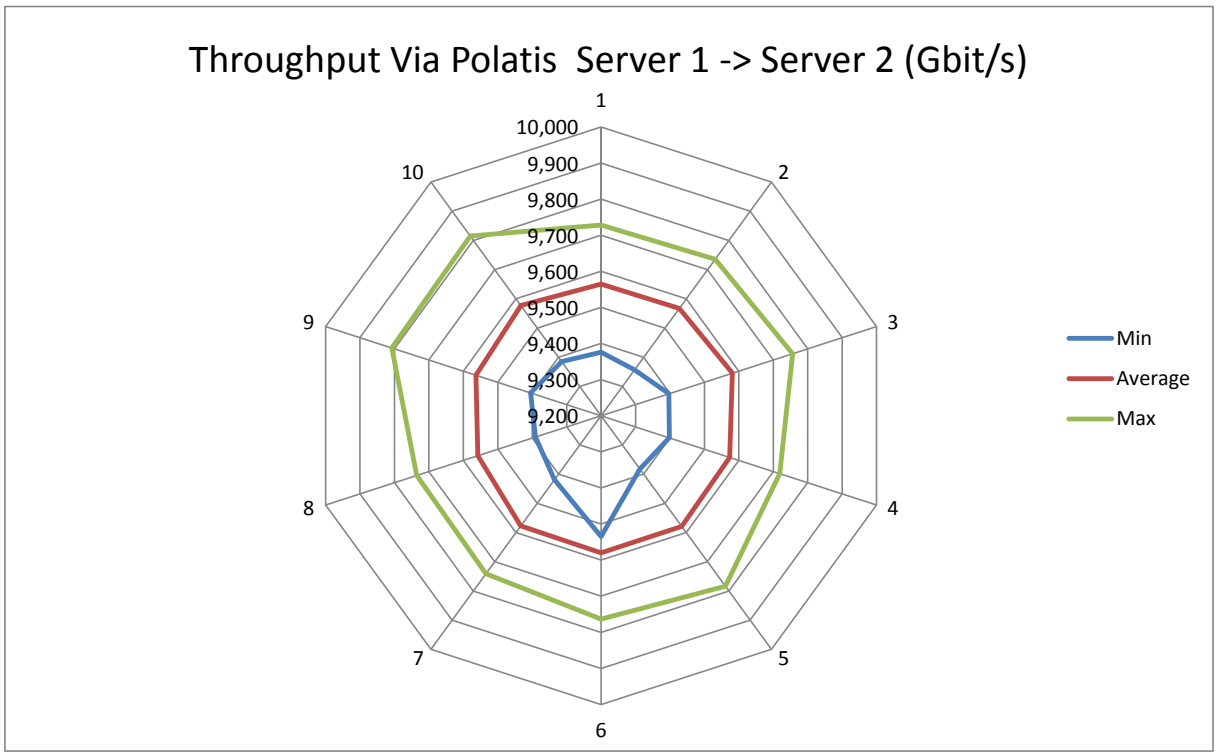


Figure 3-11: Throughput via Polatis Server 1 -> Server 2

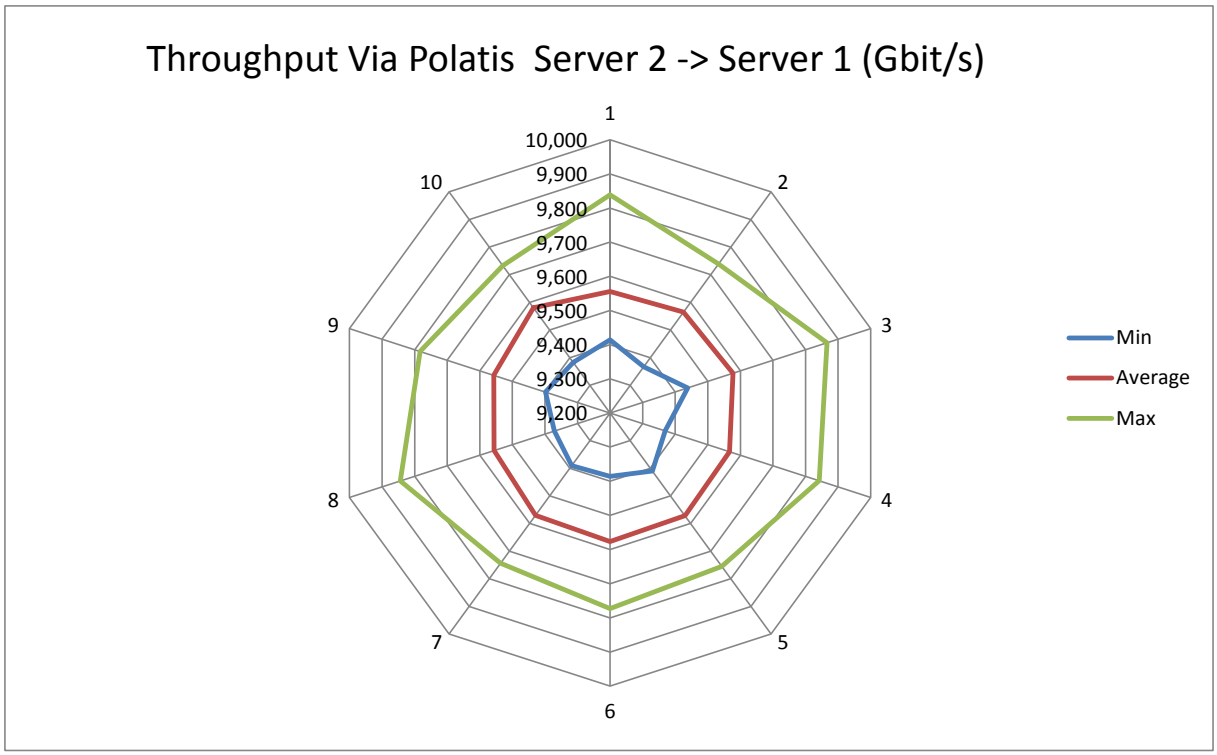


Figure 3-12: Throughput via Polatis Server 2 -> Server 1

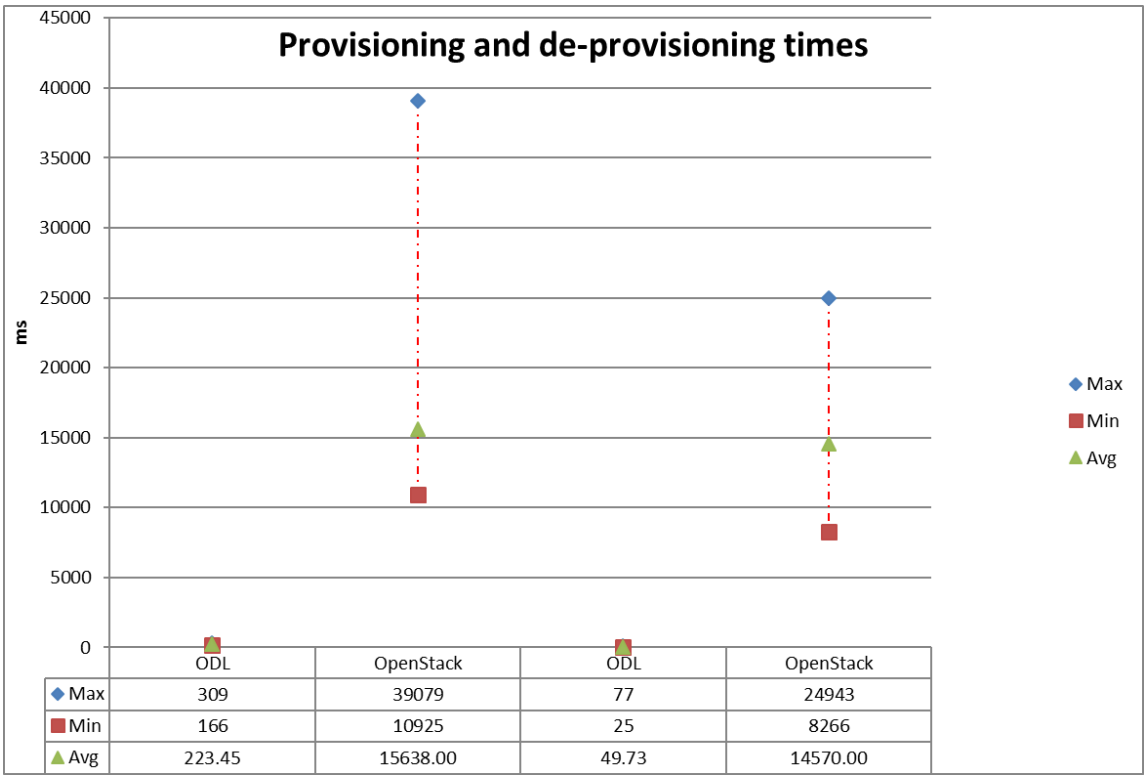
3.2.2 COSIGN VDC Service test

The test described in this section has the objective of measuring the time required to provision an instance of a simple COSIGN VDC, identifying the contribution of the different COSIGN components (orchestrator, network control plane and data plane) to the overall procedure.

The test is executed in two phases: provisioning and de-provisioning of a VDC instance. The provisioning procedure is initiated by sending a request for a VDC composed of two VMs interconnected by a virtual link, using the Horizon dashboard extended in COSIGN. This action triggers in cascade all the workflow for the VDC instantiation, including (i) the computation of the network and IT resource allocation at the orchestrator's algorithms, (ii) the interaction with the OpenDaylight controller for the configuration of the network in the compute nodes and (iii) the deployment of the VMs in each compute node. Once the VMs are instantiated, we generate traffic between them by sending ping requests; this triggers the SDN controller to initiate the setup of the DCN connections between the compute nodes so that the traffic between VMs is properly carried through the dedicated flows. The de-provisioning is requested again through the OpenStack dashboard and it triggers the removal of the VMs in OpenStack and the deletion of the connections in the DCN, managed by OpenDaylight. The test is executed 10 times and we compute average, maximum and minimum values.

In each execution, we measure the VDC provisioning time identifying the OpenStack and the OpenDaylight contributions for VM provisioning or de-provisioning time and setup or deletion of network connections respectively. Finally, we investigate more in-depth the delay introduced by the provisioning of network connections, since this constitutes the new feature proposed by COSIGN to guarantee QoS for VM traffic. The objective is to evaluate its impact on the whole VDC instantiation and the performance of the controller components in a simple scenario with a single VDC. More extensive tests on the control plane, with multiple requests, wider DCN topologies and increasing controller load are described in section 3.2.3.

3.2.2.1 VDC provisioning and de-provisioning



	ODL	OpenStack	ODL	OpenStack
Max	309	39079	77	24943
Min	166	10925	25	8266
Avg	223.45	15638.00	49.73	14570.00

Figure 3-13: Provisioning (left side) and de-provisioning (right side) time - Network (ODL) vs. computing (OpenStack) resources

Figure 3-13 shows the average, minimum and maximum values measured for the provisioning and de-provisioning time of network connections (performed by OpenDaylight) and VMs (performed by OpenStack). The duration of the procedures for network connections is in the order of few hundreds of milliseconds for provisioning and less than 100 ms for de-provisioning, while the instantiation and deletion of VMs is in the order of 15 seconds, with peaks of 40 seconds for provisioning. It should be noted that the VMs used for this test are light VMs, based on CirrOS, which require a short start-up time. We can conclude that the delay and the complexity introduced in the network side by the COSIGN architecture is negligible if compared with the overall time required for VMs instantiation and start-up.

3.2.2.2 Provisioning of network connections

In order to better evaluate the performance of the orchestrator and SDN controller components developed in COSIGN, we have analysed in detail the OpenStack and OpenDaylight logs to identify the time required to execute the different steps of the procedure for instantiating a network path in the DCN infrastructure. The workflow includes the following steps:

- Coordination of path provisioning procedures (Provisioning): time required by the Provisioning Manager in OpenDaylight to receive and elaborate the request, coordinate all the steps to retrieve the network path, update the configuration of the involved data plane devices and update the resource usage in the topology, and finally update the internal data store with the information related to the established path.
- Computation of resource allocation solution (Algorithm): time required by the orchestrator algorithms to compute and return the resource allocation solution, in terms of VM placement and network paths between hosts.

- Configuration of data plane devices (Device Configuration): time required by the SDN controller and its OpenFlow plugin to configure the required flows in the TuE and Polatis switches. This configuration is performed sending OpenFlow *flow_mod* messages (OF messages used to configure flows in the nodes) to the network nodes along the computed path. Flowmod messages for TuE switches include multiple flow rules, one for each table defined in the switch pipeline, and define traffic classifiers using OpenFlow L2 matches based on VMs MAC addresses. Messages for Polatis switches includes the details of input and output ports to create the optical cross-connection.
- Update of resource usage in the network topology (Topology Update): time required by the SDN controller Topology Manager to update the network topology with the latest information about the bandwidth availability on the links used by the established paths. The network topology must be kept updated so that the algorithms can elaborate their graphs taking into account the current network load.

As shown in Figure 3-14, the orchestrator algorithm and the OpenDaylight provisioning manager take nearly 3/4 of the entire connection setup time. Figure 3-15 shows some statistics about the absolute time for each procedure, where the X indicates the average, the horizontal line indicates the median, the border of the rectangles indicate first and third quartiles and the edges of the vertical line indicate the maximum and minimum values.

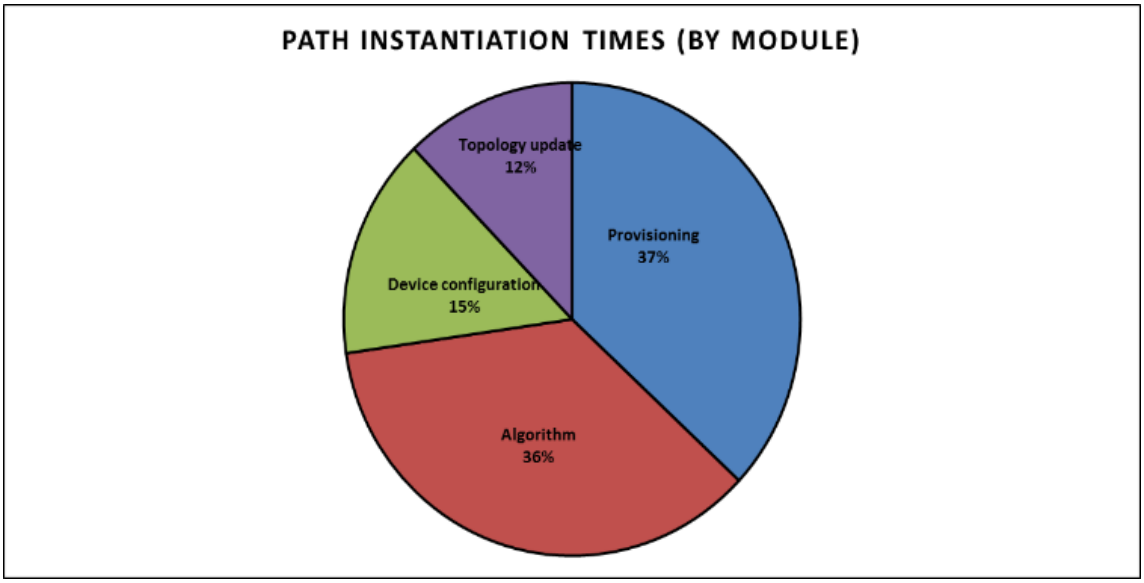


Figure 3-14: Provisioning time for a network path: time required for each workflow step (%)

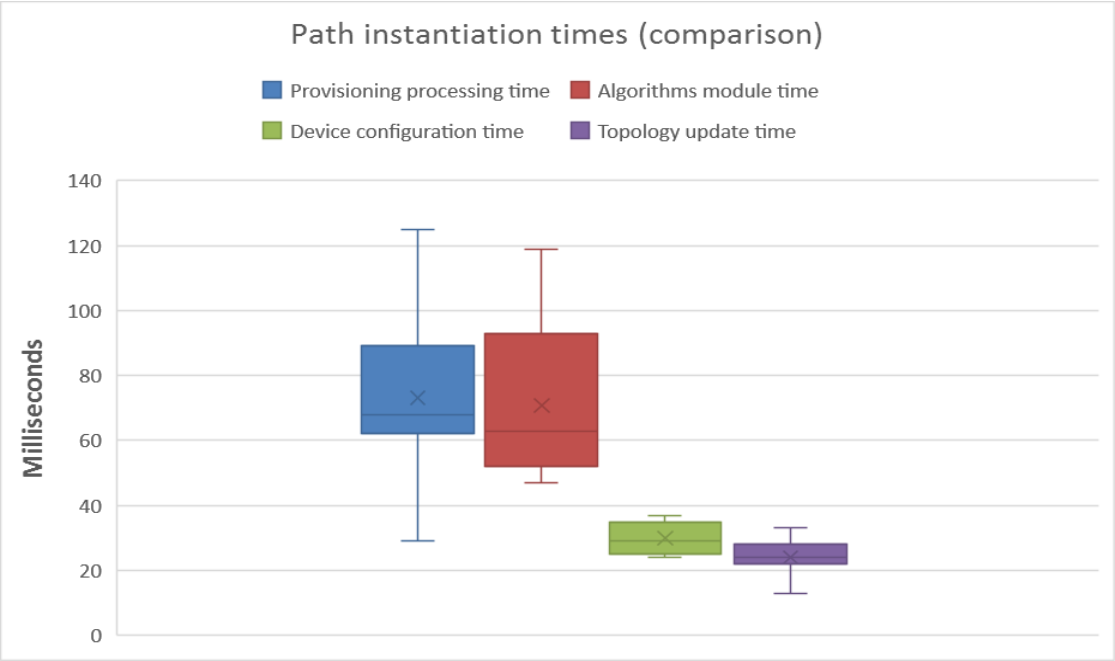


Figure 3-15: Provisioning time for a network path: absolute time required for each workflow step

3.2.3 Control plane scalability tests

In order to test the scalability of the COSIGN solution we employed a testbed based on an emulated data plane (through mininet). Only the procedures for provisioning of network paths have been tested, as the VM instantiation is obviously independent on the data plane size and topology.

The tests involved generating randomised VDC requests and sending to the SDN controller the associated network provisioning requests. They have been performed using two emulated DCN topologies of different size. We collected the computation (from request reception to the end of the path computation) and configuration (device communication for OF configuration) times, and the percentage of failed path requests due to bandwidth saturation.

3.2.3.1 Test environment

The VDC requests have a uniformly distributed size between 3 and 8 VMs, connected in a full mesh with a bandwidth of 10, 100 or 1000 Mb/s. The inter-arrival request time follows a Poisson distribution with mean 1000 ms, 800 ms, 600 ms or 400 ms, determining average VDC request rates of 60, 75, 100 and 150 requests per minute.

The first data plane topology (pertaining to Figure 3-16 (a)) is composed of 4 Polatis switches connected in a full mesh, with 6 TUE ToR (top-of-rack) switches connected to each Polatis switch and 3 hosts under each ToR switch. The other topology (pertaining to Figure 3-16 (b)) is slightly bigger, counting 5 Polatis switches, connected in a full mesh, with 7 TUE switches connected to each Polatis and 3 hosts connected to each TUE switch.

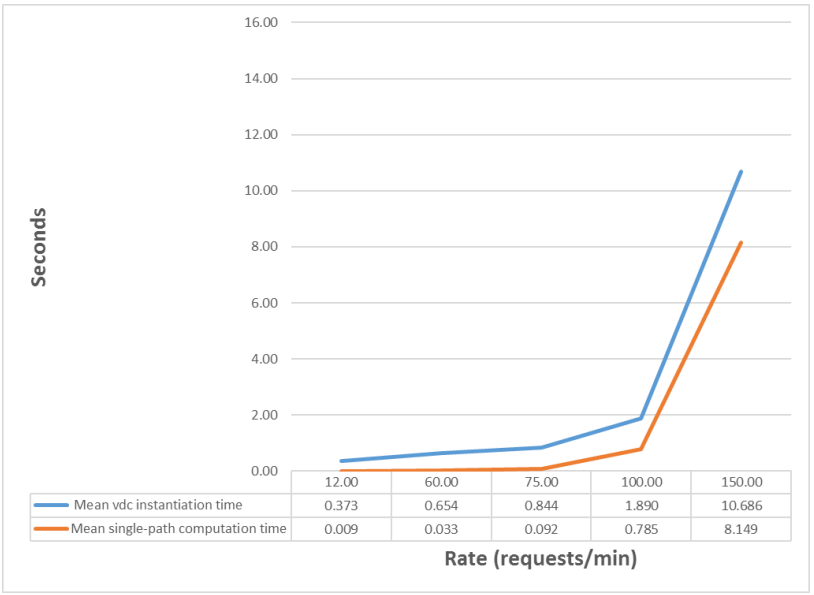
Since the objective of this test is to evaluate the performance in terms of network resource allocation only, the VM placement is computed in a disjoint manner without using the COSIGN VDC algorithms at the orchestrator and with a criterion that imposes the worst case scenario on the network side. For this reason, each VM of a single VDC is allocated on a different rack, when possible. This maximizes the number of required connections and bandwidth consumption, with up to 8 physical servers that need to be interconnected in a full mesh.

In a real VDC deployment, several VMs would be placed in the same physical server. This means that this test scenario is equivalent to configuring the underlying network connectivity to supporting

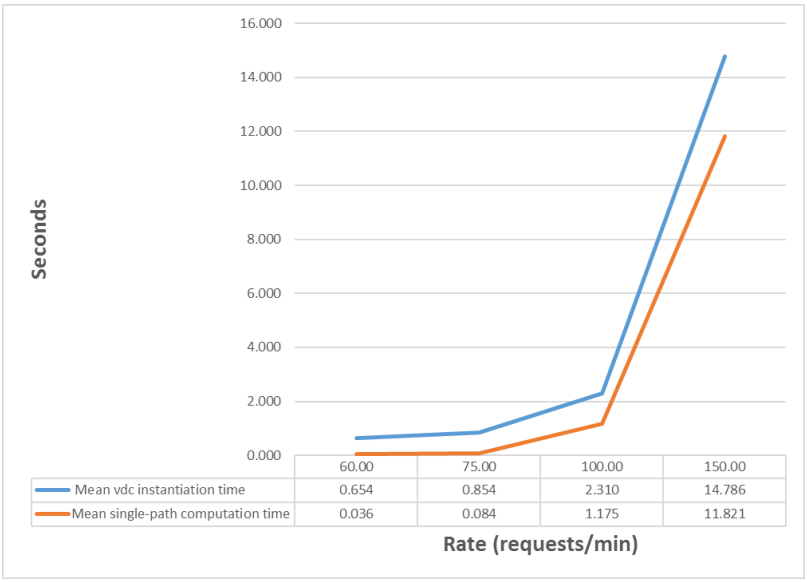
aggregated VMs traffic between physical servers for VDCs that span easily up to more than 100 VMs. This value can be considered as a very high bound even for a very resource demanding customer.

The OpenDaylight controller runs on a 2 vCPU, 4G RAM VM on the Interoute infrastructure.

3.2.3.2 Results



(a)

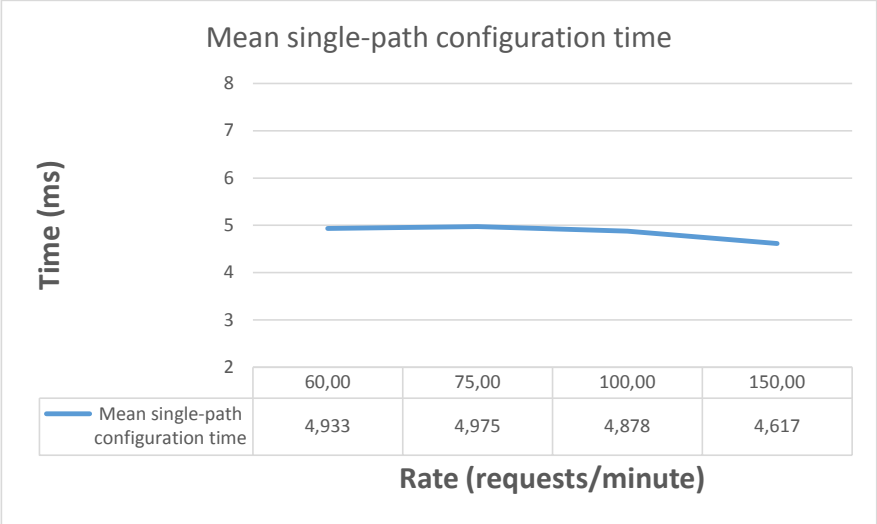


(b)

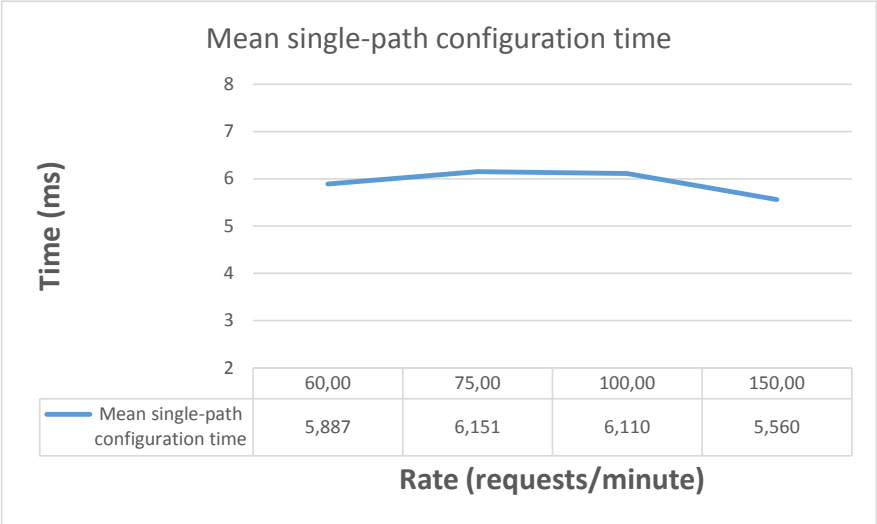
Figure 3-16: Scalability tests results – (a) first topology, (b) second topology

The results in Figure 3-16 (a) and (b) show that the VDC and path instantiation times are below the 2sec mark on both topologies up to the request rate of 100 VDC requests/min. On top of that, the VDC instantiation times stay very close to the baseline time of ~0.4s (obtained in a control experiment with a minimal rate of 12 requests/min) for all rates under 75 requests/min. This is also compatible with the instantiation times registered during the data plane tests, which involved just a few connection requests. As the request rate rises above 100/min, however, the instantiation time rises rapidly to tens of seconds per VDC. Not shown above are the results for tests run with even higher rates (180 requests/min and higher) as the controller stopped being able to serve requests at all, and the test failed to complete. Regarding the network size, the times are almost equal up to the rate of 75 requests/min, and higher by about 20% with a rate of 100 requests/min.

The chart below (Figure 3-17) shows the configuration (i.e. communication with devices) time, which is negligible at all rates, staying around the 5 ms mark. This time is much lower than the same time in the data plane tests since the controller here is communicating with local mininet OpenVSwitches. The time slightly increases on the second topology, probably due to the higher number of device communication channels it has to keep open and the overhead this causes on the southbound OpenFlow communication plugin.



(a)



(b)

Figure 3-17: Device configuration time – (a) first topology, (b) second topology

3.3 Comparison current VDC and COSIGN VDC

The VDC service provided by Interoute supports two different models. The first is based on the full automation of the provisioning procedure and delivers VDC instances on-demand and with a customizable number of VMs, but without guaranteeing any QoS for the VM traffic. In this case, the DC network is statically pre-configured and the network resources are shared among the different VDC instances, resulting in best effort traffic. The second VDC model is dedicated to "gold" customers who need an advanced VDC configuration and, in particular, guarantees for the VM traffic. In this case, the VDC procedure is not automated and it requires the manual configuration of the network environment in order to establish dedicated connections with the requested level of QoS.

COSIGN brings a major benefit for this kind of procedure, since the integration between orchestrator and SDN controller allows to apply the on-demand provisioning approach to the advanced VDC

model and automates the DCN configuration for QoS guarantees as part of the standard workflow of VDC provisioning. This evolution reduces not only the VDC service costs (limiting the need for the manual intervention of the DC administrative operators), but also the delivery time for complex VDCs. In particular, the customer can request their own advanced VDC instance, specifying the required QoS constraints, using a self-service portal (something similar to the COSIGN VDC dashboard developed in the project) and receive access to the VDC in minutes, as for simple "best-effort" VDC instances today. In fact, from the analysis of the previous test results, we can easily deduce that the additional delay introduced by COSIGN for the integrated network connection provisioning is in the order of few seconds, even in conditions of high loads, high request rates and high VDC size. This value is much less than the provisioning time requested to instantiate even a single VM (which is in the order of minutes for VMs with a reasonable set of installed applications, as usually requested in VDC services), and negligible in the whole provisioning time of typical VDCs, that usually include multiple VMs.

In terms of performance and scalability, the solution proposed for the network control is applicable with low delay (around 2 seconds) for up to 100 VDC requests per minute. This can be considered a very high request rate for the VDC service, since it would mean to receive more than 1.5 requests per second, from different customers who want to continuously create new VDC instances. However, the network provisioning time still remains in the order of 10-15 seconds, so still much less than 1 minute, even up to 150 VDC requests per minute. At the rate of 100 VDC requests per minute, the variability introduced by increasing the DCN topology size is negligible.

The control plane failure point due to overloading is measured around 180 VDC requests per minute for time periods of several minutes (i.e. 3 VDC requests per second continuously issued for more than 15 minutes). This can be considered as a clear condition of Denial of Service attack and it can be easily prevented by the usual Intrusion Detection and Prevention Systems employed in IRT infrastructures. Thus the COSIGN solution is valid and fully applicable in the context of advanced VDC services.

The second benefit introduced by COSIGN is related to the usage of the optical infrastructure, which increases the available DCN bandwidth and guarantees low latencies, as verified in the data plane tests. This is a great advantage for DC operators, since they can serve more customers on the same shared infrastructure without decreasing the performance of the single services, but also for the customers who may exploit the DCN low latency to run cloud applications with stringent QoS requirements.

4 Final Demonstrators: Large scale Mid-term and Long-term scenarios

4.1 Scenario 1: Large scale mid-term data plane

4.1.1 Hypercube structure for datacenters with and without optical shortcut

One of the central achievements of the COSIGN datacenter architecture is to include optical switching functionalities. In the demo, 8 commercially available HP Aruba switches are connected in a Hypercube structure. Each of the Aruba switches also has connectivity to the Polatis switch, adding the optical shortcut functionality. This can be used to increase the robustness and reliability of the cube structure in case of link failures, apply load balancing mechanisms in case of high traffic loads, as well as to allow for shortcut connectivity for latency sensitive applications.

The advantages of being able to re-route traffic in a Hypercube using the Polatis switch are many fold:

- Load balancing: In case some of the links in the Hypercube structure are being overloaded, the problem can be alleviated from the suffering links by redistributing traffic via the Polatis switch.
- Prioritization: The shortcut through the Hypercube allows to prioritize certain traffic within the Hypercube to be routed on the best possible links. This could mean allowing for shortcuts through the Hypercube structure via the Polatis switch for latency critical applications.
- Robustness/Reliability: The shortcut links via the Polatis switch also increase the robustness of the Hypercube structure. It means that for each single link failure, the cube provides inherent protection by being able to circumvent failed links via the Polatis shortcut, similar to local span protection.

4.1.1.1 Latency measurements in Hypercube with and without Optical Shortcut - Single Cube

The first test illustrates the gain in latency using the optical shortcut through the Hypercube via the Polatis switch under varying load conditions. The switches are logically configured as illustrated in Figure 4-1. We use a Xena Appliance tester to measure end to end latency on the maximum Hypercube distance compared to the Polatis optical shortcut. The Xena testers are used to load the Hypercube with traffic of varying load, i.e. varying percentage of a 10 Gpbs connection. Packet size is kept fixed at 64 byte.

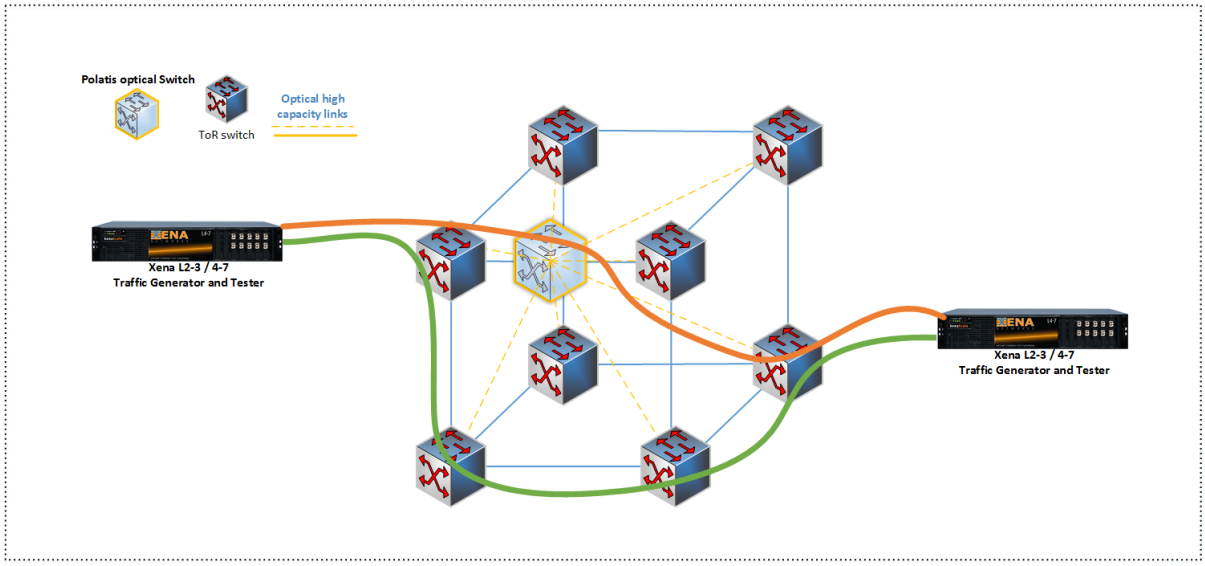


Figure 4-1: Scenario with 8 HP Aruba switches and Polatis switch configured in a Hypercube. Xena tester for traffic generation and measurements.

The measurement results are illustrated in Figure 4-2. Measurements reveal a latency reduction of approx. 3.5 microseconds when using the optical shortcut compared to the baseline scenario of using only the switches in the regular Hypercube.

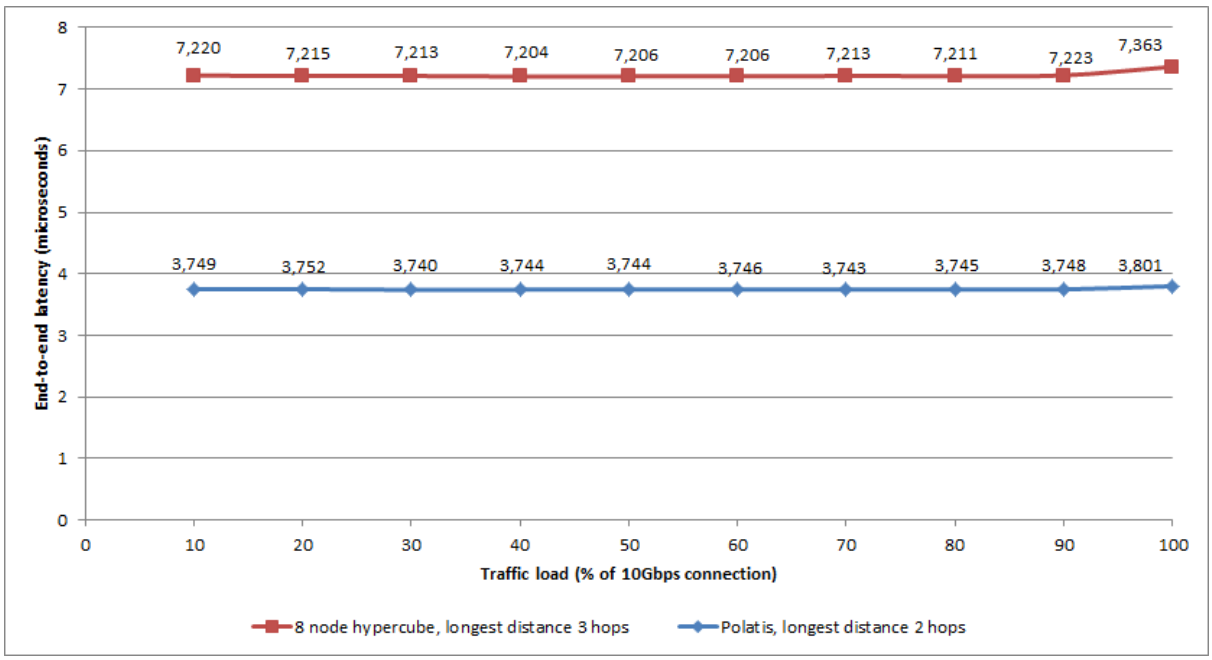


Figure 4-2: End to end latency measurements comparing longest distance in 8 node Hypercube to a Hypercube with an optical shortcut through the Polatis switch

4.1.1.2 Latency measurements in Hypercube with and without Optical Shortcut - Double Cube

The effect of the Polatis optical shortcut is expected to become even more prominent when the Hypercube is scaled to 16 nodes, which is conceptually illustrated in Figure 4-3. Here, one Polatis switch is shared between a double Hypercube structure.

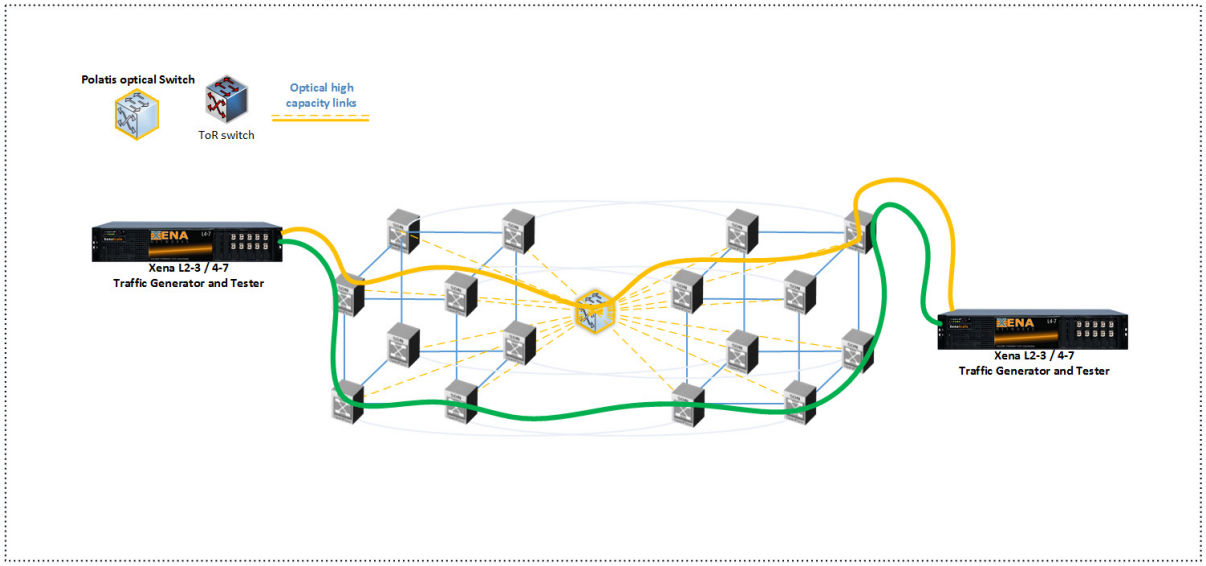


Figure 4-3: Conceptual scenario of 16 HP Aruba switches and Polatis switch configured in a Hypercube. Xena tester for traffic generation and measurements.

Measurements comparing a longest distance of 6 hops via the double cube and via the Polatis optical shortcut are shown in Figure 4-4. Results show that latency can be significantly reduced also in the double cube structure, of approx. in the range of 8.8 microseconds.

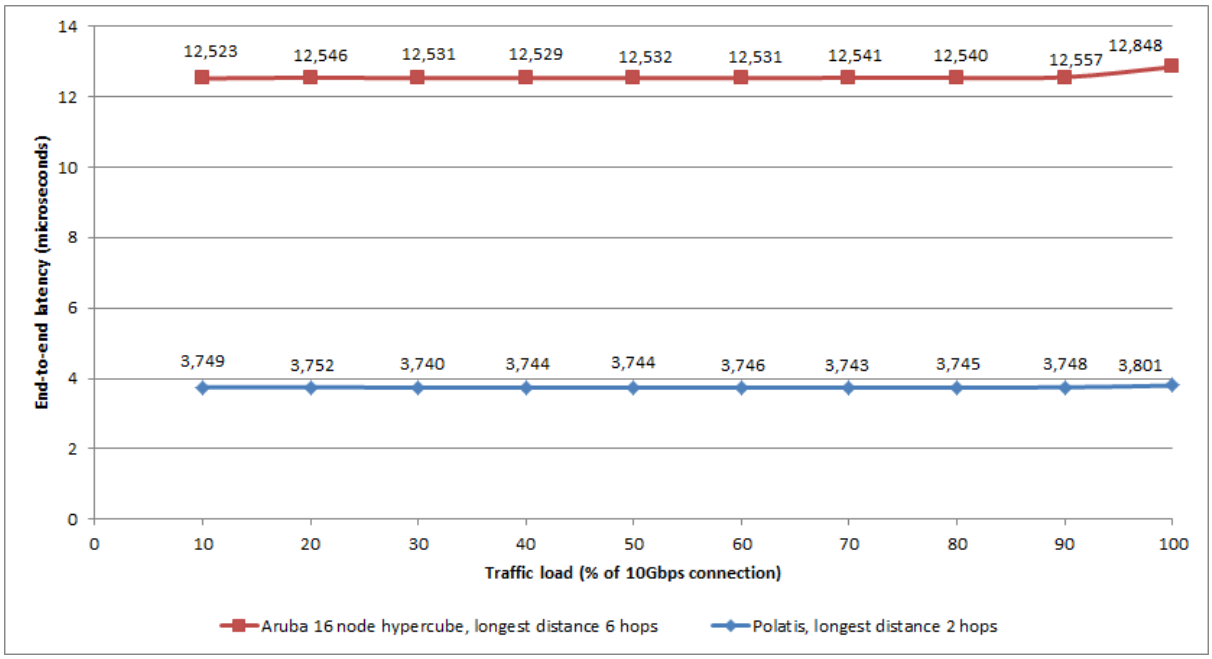


Figure 4-4 End to end latency measurements comparing longest distance on 16 node Hypercube to a Hypercube with an optical shortcut through the Polatis switch

From the scaling studies and latency measurements it can be derived that the optical shortcut can decrease latency in the cube structure significantly, and it will become even more prominent if the cube size is increased. However, this will require that all TORs have direct connectivity to the optical shortcut switch. In an ultra large scale scenario, the connectivity may have to be reduced to fewer connection points.

4.1.2 Combination of simulation and real equipment for datacenter performance evaluation

4.1.2.1 Proof of concept study

In order to facilitate the performance analysis and scaling studies of datacenters, we have integrated simulation and real hardware. The results of the first proof of concept study were published and presented at the ONDM conference May 2017 [5]. We use Riverbed Modeler's System-in-the-Loop (SITL) tool [6], which supports required functional capabilities to enable this software-hardware integration. The tool provides the linkage between the "real world" and the simulation that is running inside a computer. Any type of device, e.g., a router or a switch, can be linked to the simulation via the workstation's Ethernet port. Let's assume a packet that is generated by a real server, processed in a simulated datacenter network, and terminated by another real server. Once the packet is generated on the real server, it will be transmitted via Ethernet to the workstation running the simulation. Most importantly, the real time and the simulation time are running continuously, ensuring that timing is kept consistently throughout the entire system. The real packet is then converted into a simulation packet that will then be processed throughout the simulation. When the packet has passed through the desired simulation path, it is mapped at the external interface and converted back to a real packet. The real packet is then sent from the workstation through the Ethernet interface towards the real equipment. A simplified view of packet translation procedure is illustrated in Figure 4-5.

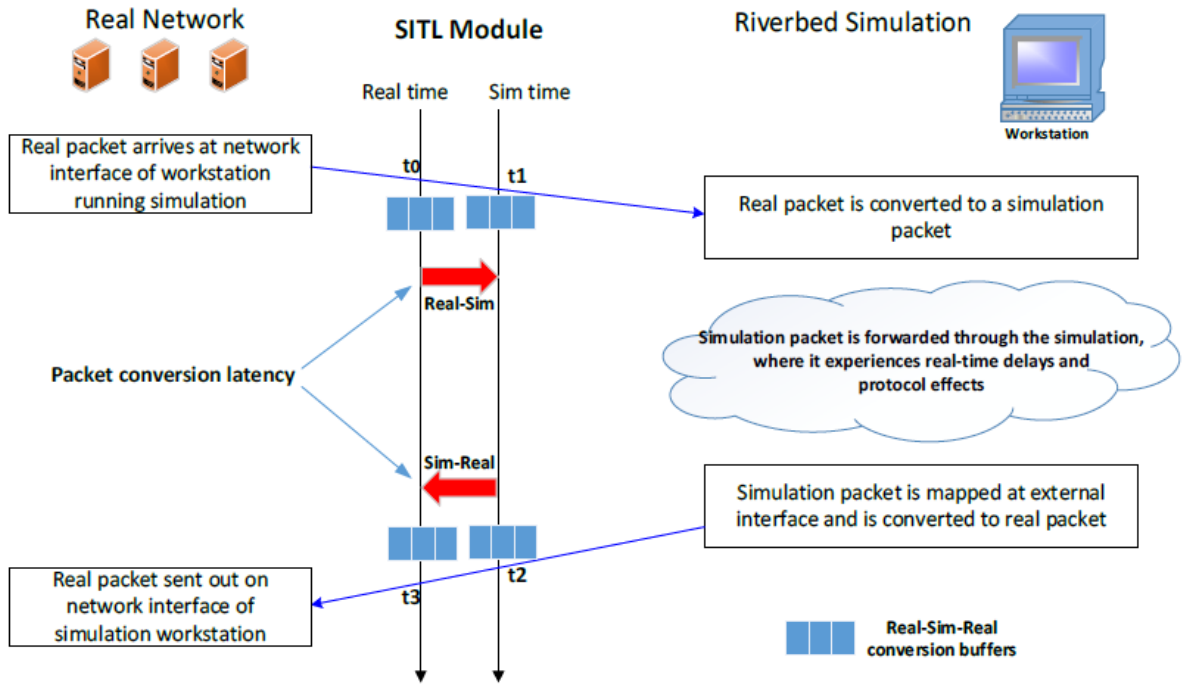


Figure 4-5: Packet flow between real and simulation equipment

In this work, the Ring-of-Rings [8] (as presented in D1.4) and the HyperCube [7][9] architectures are analysed, see Figure 4-6.

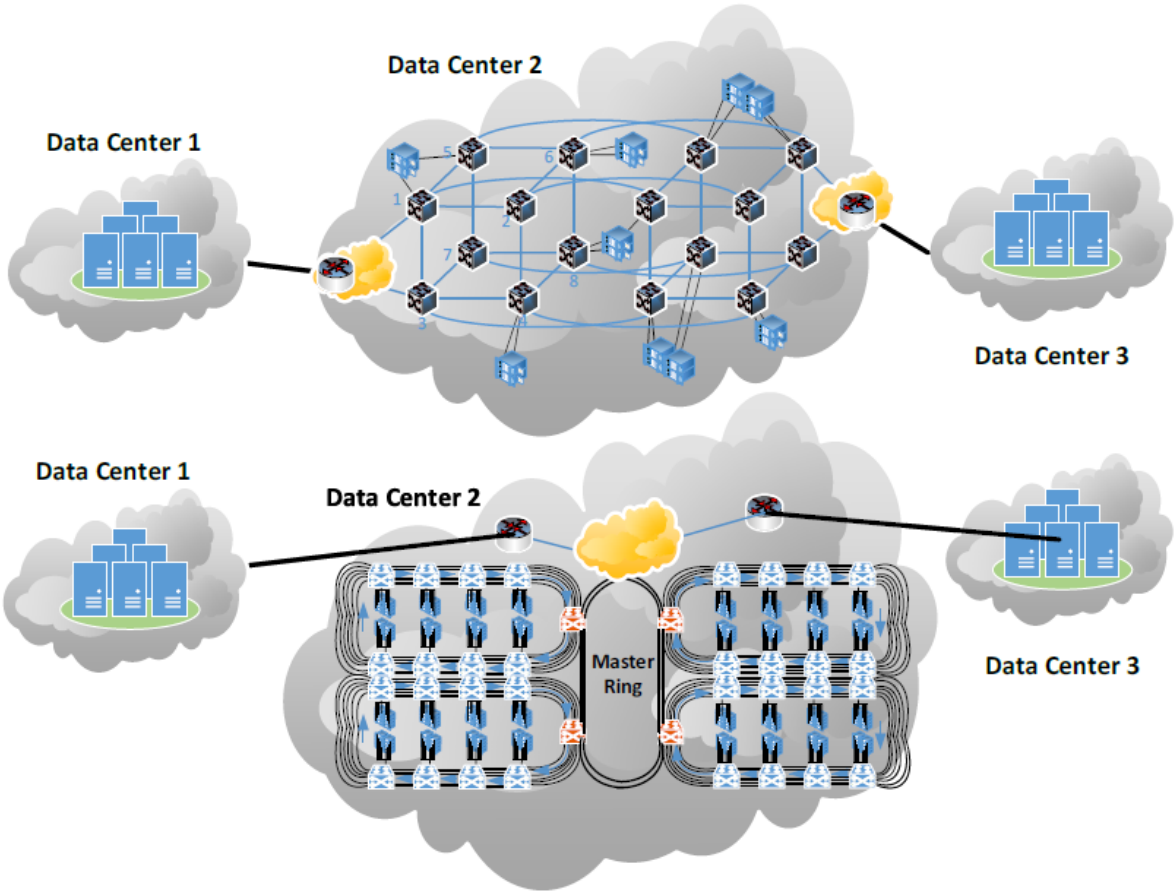


Figure 4-6: HyperCube and Ring-of-Rings architectures

We are using two state of the art traffic generators, namely Xena Testers [11], to generate traffic on different layers between Datacenter 1 and Datacenter 3, cf. Figure 4-6. The traffic is passed through Datacenter 2, which is modelled using either a HyperCube or Ring-of-Rings internal architecture.

Every topology is composed of a multitude of TOR switches, each of them having multiple servers attached. Thus, the architectures are built as simulation models to illustrate scalability without having to acquire multiple TOR switches.

The packet translation efficiency is highly dependent on the translation level needed (see Figure 4-7). Some traffic may not be terminated within the DCN, hence certain payload types will be just copied as a block of bits, not touching the corresponding headers, resulting in faster translation. The simulation configuration (setup) and real hardware is shown in Figure 4-8, where the simulation model and real equipment are linked via the virtual SITL gateways (software modules) [10].

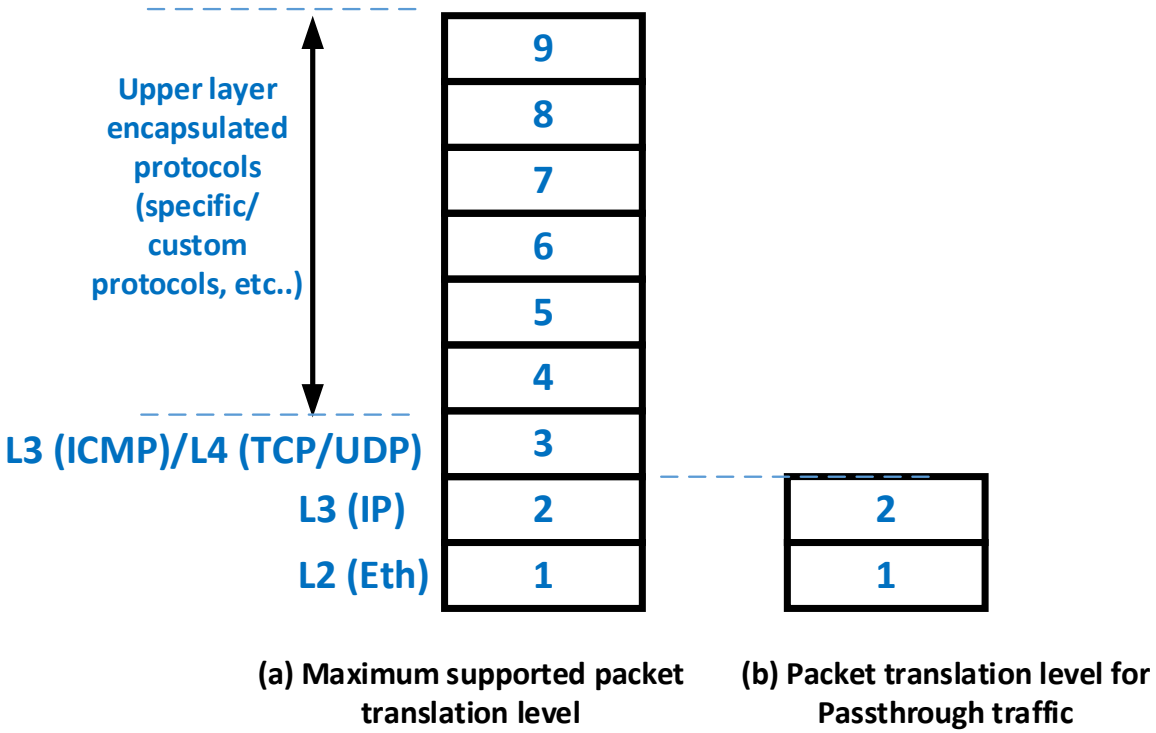


Figure 4-7: Packet translation depth (level) at the real/simulated interface

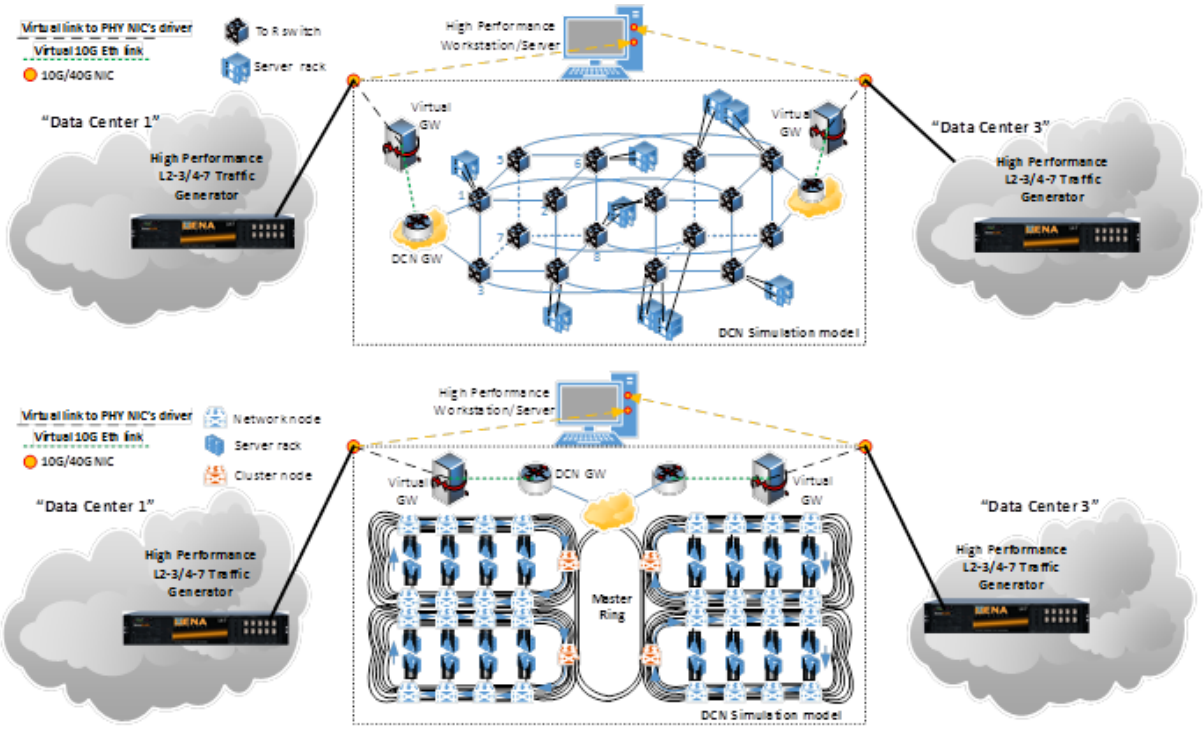


Figure 4-8: Experimental setup combining real equipment and simulation

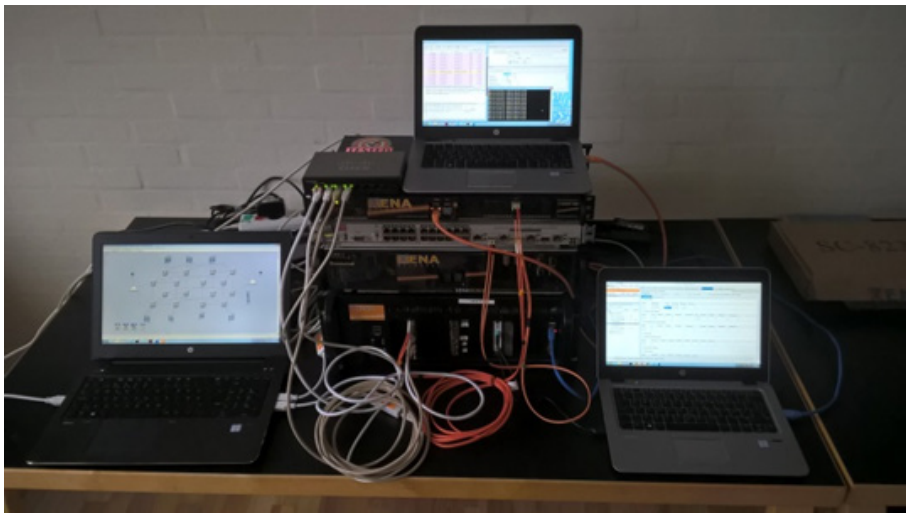


Figure 4-9: Picture of the experimental setup

The entire system setup is shown in Figure 4-9. Note that in the depicted experimental setup we used portable computers for the initial demonstration tests. In this case a maximum rate of 80 and 120 Mbit/s was achieved in our tests for the ICMP and TCP traffic, respectively, on a 1GE network interface due to encountered Windows socket buffer overflow (operations on non-blocking sockets that cannot be completed) as described in [12]. Large scale experiments will run on a dedicated set of more powerful servers. The "workstation" on the left (black) is running the simulation model of datacenter 2. The Xena testers in the middle are emulating datacenter 1 and datacenter 3 by generating different predefined traffic patterns.

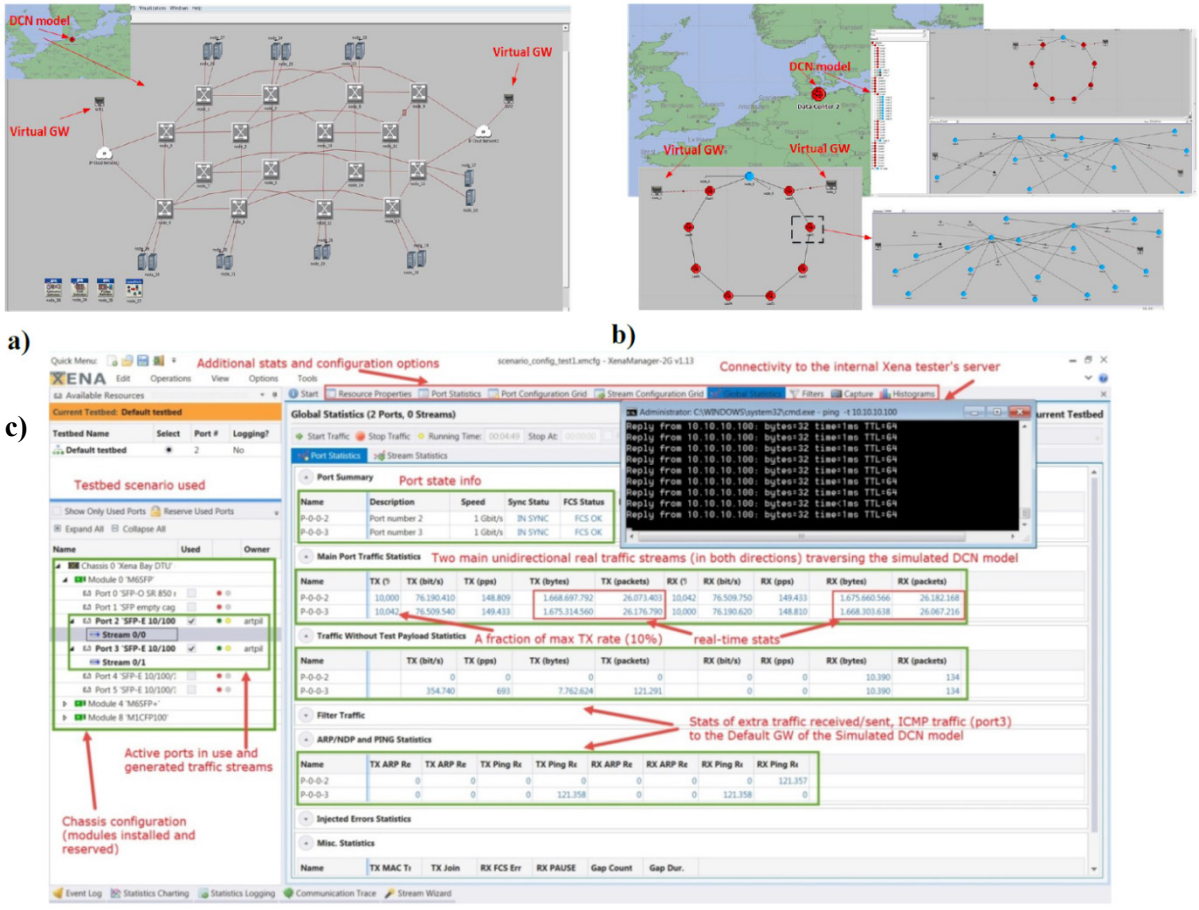


Figure 4-10: Software components of the testbed setup: a) Simulation model of HyperCube-16 architecture; b) Simulation model of Ring-of-Rings architecture; c) Picture of Xena L2-3 tester GUI

The main benefits of using these high performance testers are: a) the possibility to generate realistic application layer communication patterns (*pcap*-based replay); b) the possibility to emulate millions (when we need scale) of concurrent traffic flows by utilizing multiple available transceiver modules (1GbE, 10GbE, 40GbE and 100GbE rates, depending on the modules used). For a realistic experimental case study it is sufficient enough to have a few 10G or 40G modules, since present day workstations or servers can be equipped with 10G or even 40G NICs (Network Interface Card) to provide a reasonable uplink/downlink for our modelled DCN.

The laptop on top of the system is analysing packets by running a Wireshark tool. The software for the Xena testers is executed on the laptop on the right side of the picture.

Detailed screenshots from the individual computers and models can be seen in Figure 4-10, showing the simulation models and Xena tester software interface, respectively.

It is important to point out that evaluation of the packet translation latency parameter at the real-simulated network interface is crucial in the context of DCN performance analysis, because of much more stringent timing requirements, compared to conventional networks. In modern (and future) DCN environments there are several critical factors which set DCNs aside into a different "networking" category, namely ultra-low latency and high throughput requirements, ultra-short duration of the vast majority of intra-DC traffic flows (in the order of a few 10s or 100s of *ms*), and significantly larger east-west (internal, remaining within a DC) traffic volumes as compared to south-north (external) traffic.

In terms of results, one of the most important measures in datacenter networks is latency. We thus measure the time that it takes to traverse the SITL gateway nodes in both directions, namely the conversion delay on this virtual interface. This parameter is of paramount importance when it comes to the further performance evaluation of the simulated DCN topology in terms of delay, because it directly affects the accuracy of the obtained measurement results.

The stress-testing was performed by loading the simulated transit) DCN, datacenter 2, with a large number of high bit-rate ping flows (symmetric bidirectional traffic). The results are illustrated in Figure 4-11. We observe that under the load of ~ 75-80 *Mbit/s* (see Figure 4-11(a)), the time-average conversion delay for the incoming traffic (real-to-simulated) is fluctuating around 2.0 μs per packet, while in the opposite direction (simulated-to-real) it is almost 60% higher (Figure 4-11(b)). However, this is a relative difference, since this result shows a time-averaged value. It is more interesting to look at the Cumulative Distribution Function (CDF) of this parameter, being a better indicator. As it can be seen in Figure 4-11(*c*), for around 95% of collected samples, the per packet conversion delay is less than 6 μs on average for both directions, with the worst case scenario being below 10 μs. As a result, this penalty must be taken into consideration while evaluating the performance metrics of a DCN under consideration. The evolution of the queue size in virtual SITL gateway is shown (Figure 4-11(*d)*) to be relatively low (3-8 packets on average), but this parameter is very important, since it will affect the queueing delays under much higher traffic loads.

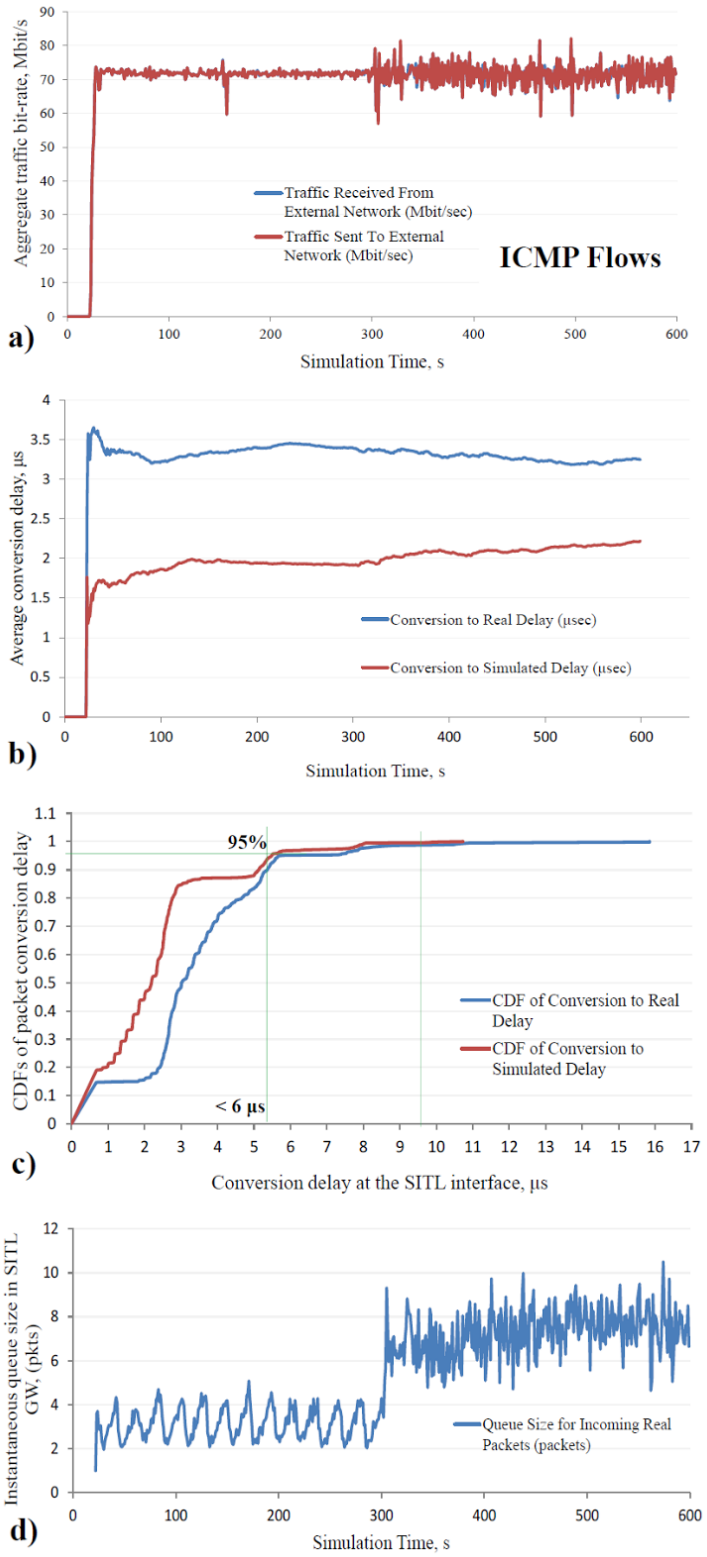


Figure 4-11: Results of Simulation model's stress-testing. Packet conversion efficiency at the SITL GW interface. Test 1

In another test, the SITL gateways were loaded with asymmetric traffic flow, namely by emulating a transmission of a large data file (3.41 *GB*) via TCP connections (data in the forward direction, streams of ACKs in the reverse) with the average data rate of 120 *Mbit/s*. A bursty traffic generation profile was configured for the data transfer experiment. We analysed the CDFs of bidirectional packet conversion and queueing delays, presented in Figure 4-12 (*a* – CDF for the TCP ACK flow, *b* – Data

traffic flow, *c* – queueing delays for the incoming real packets). As can be seen, statistically, the conversion delay is in the order of a few *μs* in both directions, and 95-percentile latency is in the same range for both flows. However, considering the proportion of packets corresponding to each flow (Data and TCP ACKs), the average number of Data packets per second (pps) was around 14000, whereas for the TCP ACK stream it was ~ 50% of that value, namely around 7000 pps. This proportionality is expected, since by default TCP connections were using a *delayed ACK* mechanism. Thus, this dependency is reflected in the conversion delay results in Figure 4-12(*a*) and (*b*), where the conversion delay of the 50% of the TCP ACK packets is less than 1 μs, while 50-percentile of the data packets experience delays twice as large (~ 2 μs).

When it comes to the packet queueing in SITL, Figure 4-12(*c)* shows the delay experienced by more than 92% of ACK stream packets is under 1 *ms* (at the order of a few *μs*), while data packets are queued up to 4 *ms* (95-percentile), with the worst case scenario of ~ 10 *ms*. The latter values are relatively high and will thus impact the performance results.

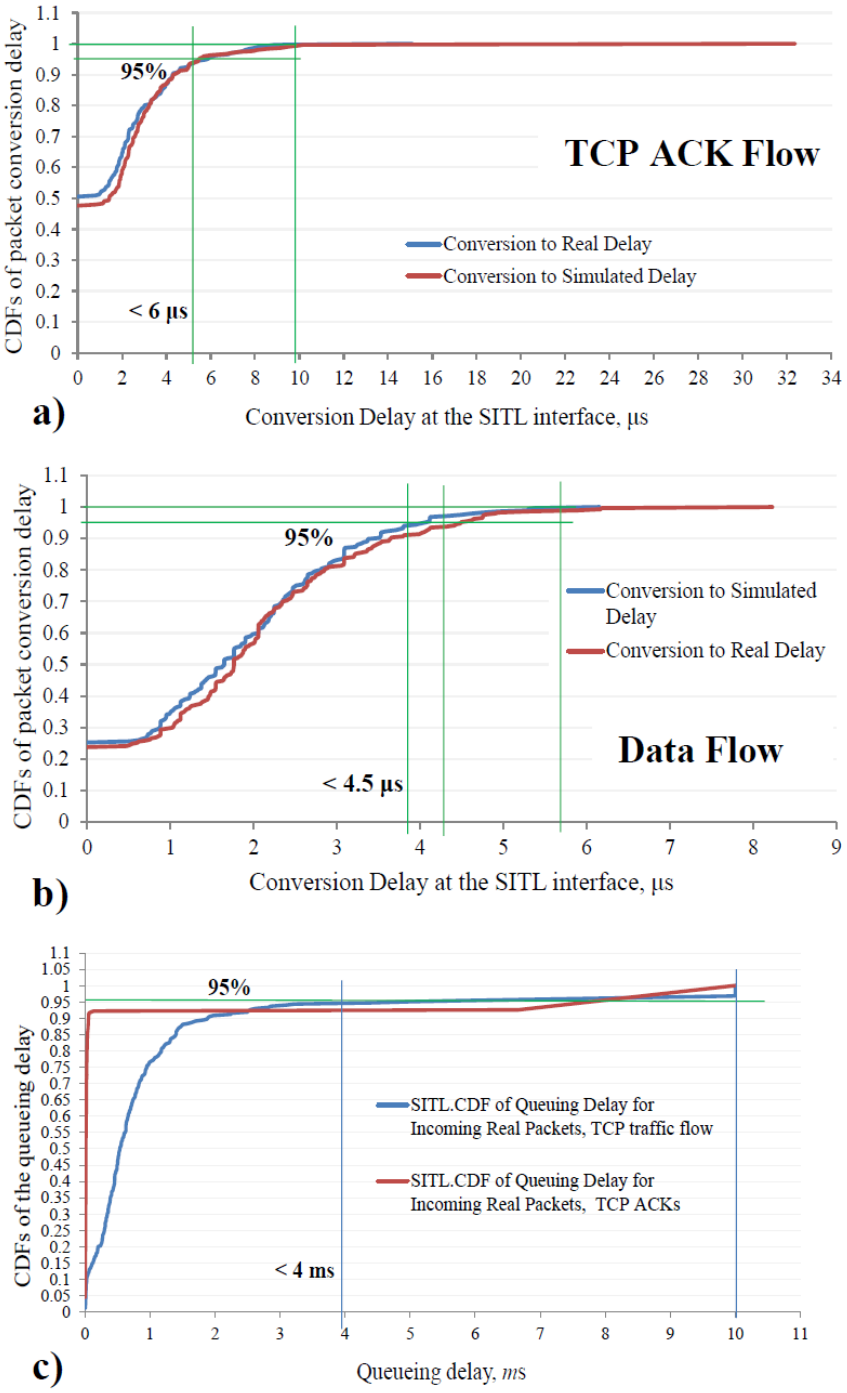


Figure 4-12: Results of Simulation model's stress-testing. Packet conversion efficiency at the SITL GW. Test 2

We evaluated the dependency of the packet flow rate on the average conversion delay by statistically sampling the packet rates and corresponding (by simulation timestamp) conversion latency using the obtained distributions (Figure 4-13 *a, b*). The preliminary results (see Figure 4-13 *c*) show that there is no clear link between the packet rate and conversion delay incurred, and the stochastic nature may be a result of several other factors, such as specifics of packet capture by the WinPcap [13] (libPcap for Linux) module, implementation of the conversion functions (code) and characteristics of the NIC installed (buffering, protocol checksum offload, etc.).

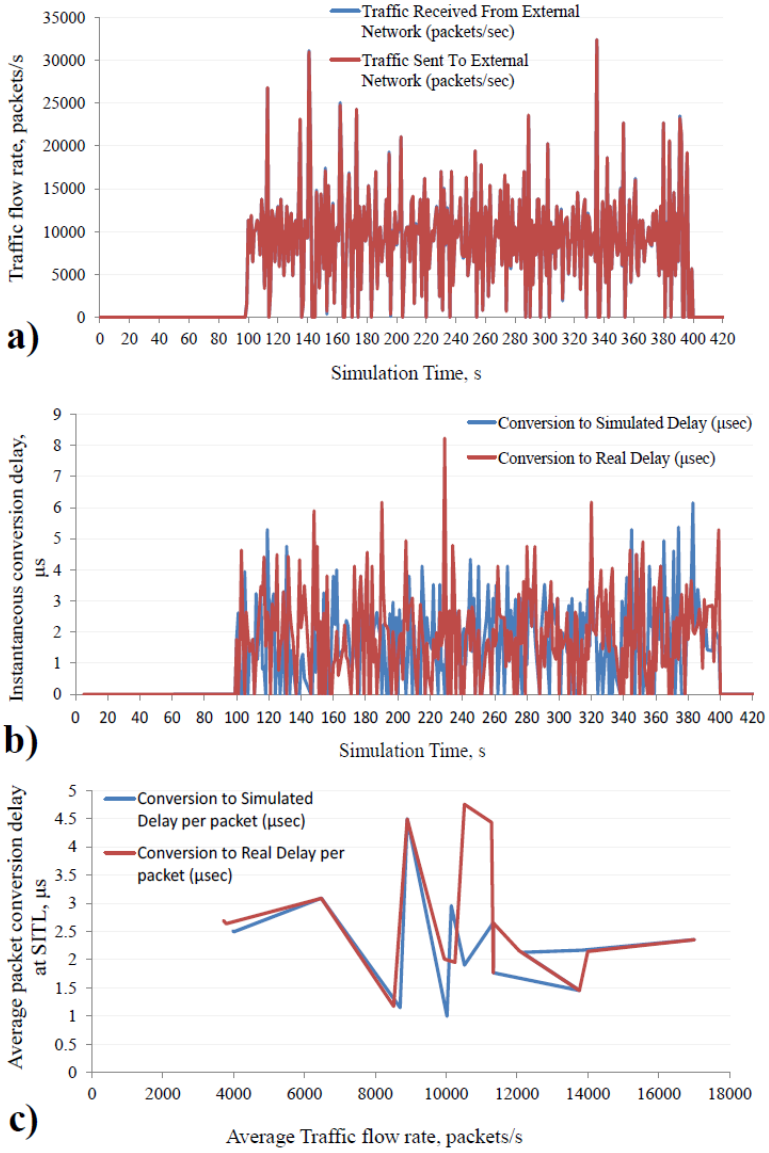


Figure 4-13: Dependency of the packet conversion latency on the traffic flow rate in virtual SITL gateway node

Summing up, in this test scenario, we have detailed an approach for combining real hardware and simulation for the purpose of evaluating the performance and scalability of datacenter networks. We described how the Riverbed System-in-the-Loop (SITL) tool can be used to interconnect real world and simulation under continuous timing constraints, without having to invest in vast amounts of expensive hardware. Our results show that the SITL gateway adds a conversion delay in the order of microseconds as well as load-dependent buffering delays that must be taken into consideration for the analysis of latency measurements.

4.1.2.2 SITL Setup interconnecting commercial Aruba Switches and Polatis

In Figure 4-14, the scenario of combining SITL simulation with real switching equipment and testers is illustrated. Traffic originates from the Xena Tester, and via a SITL gateway it is translated to the simulation world and can thus be passed through any desired datacenter structure. Afterwards, traffic is converted into the real world again, and is then routed through real equipment. Further SITL gateways allow adding larger virtually simulated datacenters, before traffic is re-enters the real world and is terminated at the testers.

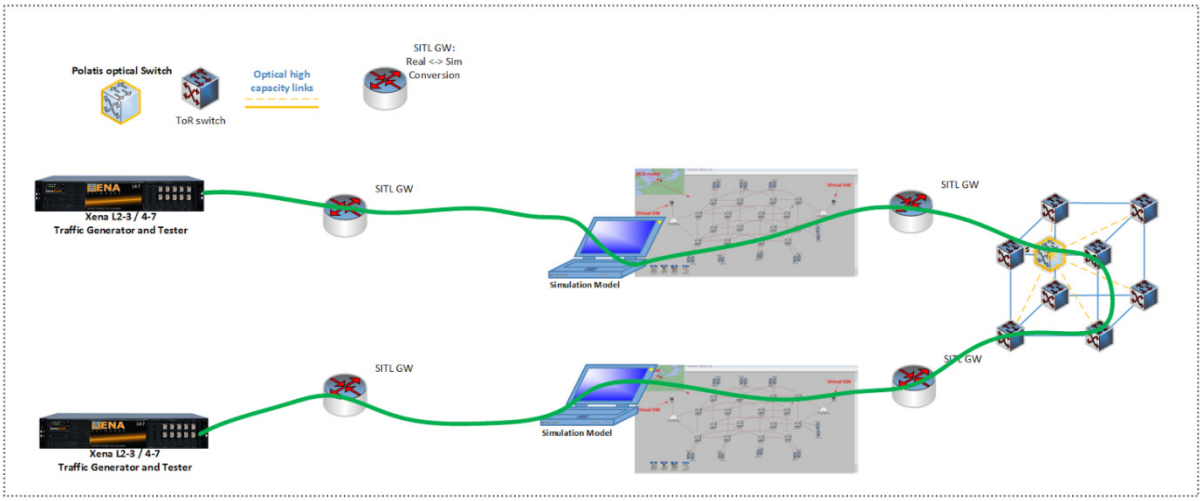


Figure 4-14: SITL setup combining simulation and commercial switches

The following figures exemplify simulation models designed using Riverbed Modeller tool with SITL functionality having in mind the following research objectives:

- Figure 4-15 illustrates an 8-switch Hypercube model with connectivity to Xena testers and an external SDN controller. This model can be used for an extended POC (Proof-Of-Concept) test (integration of the simulated SDN Datacenter setup with real devices and a controller).
- Figure 4-16 depicts an integration model, consisting of a simulated 8-node Hypercube, interconnected with the physical 8-node Hypercube setup and a Polatis OCS. Thus, a powerful server machine is equipped with up to 16 electrical-optical interfaces to support these connectivity demands. The benefit of such a scenario is to evaluate different traffic forwarding strategies in a full symmetric 16-hypercube structure (50% simulated, 50% real nodes).
- Figure 4-17 shows a model, designed for scalability studies, with a primary focus on asymmetric (incomplete) hypercube structures, deemed to be much more efficient from the scalability point of view, as compared to the symmetric structures. Hence, a combination of a 44-node simulated hypercube is analyzed in combination with a subset of real SDN-enabled commercial switches, Xena testers and an SDN controller.

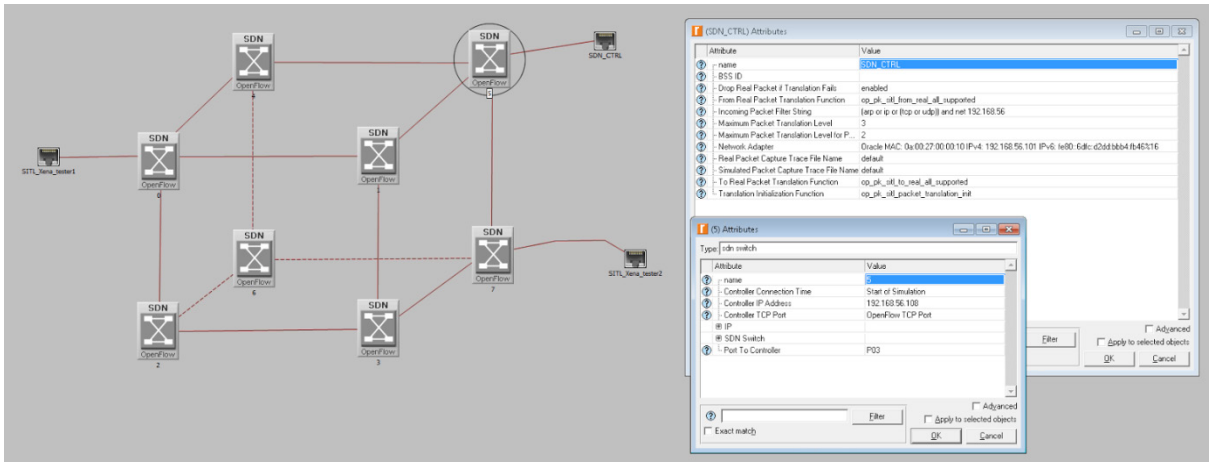


Figure 4-15: SITL model 2: 8-hypercube model

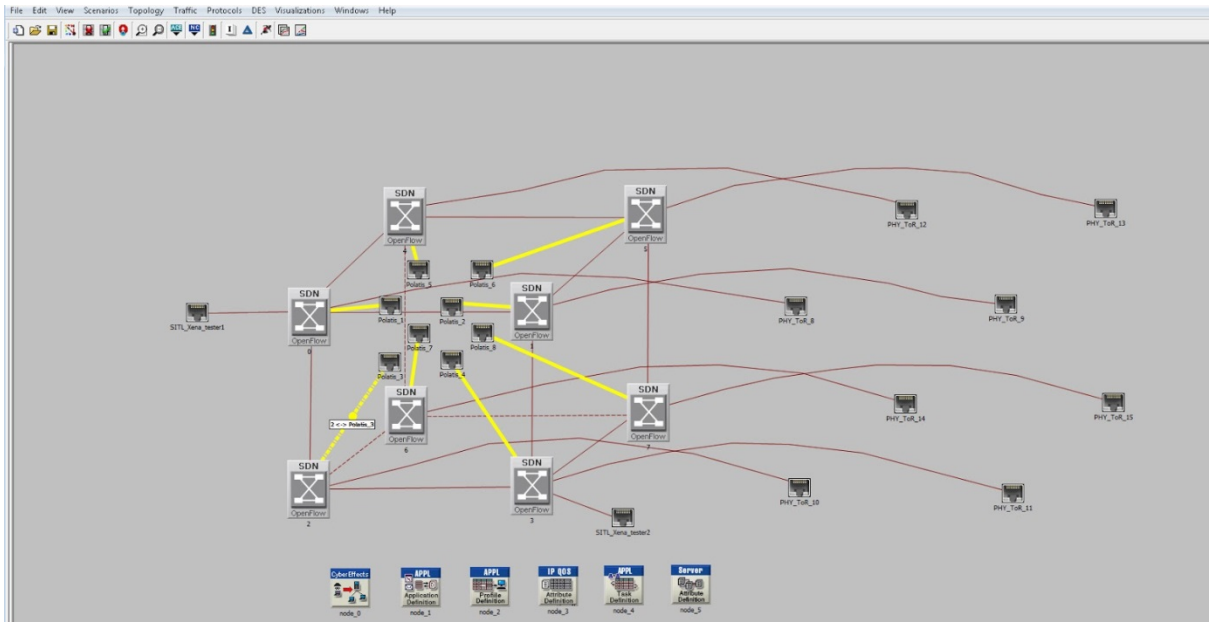


Figure 4-16: Connectivity to Polatis switch

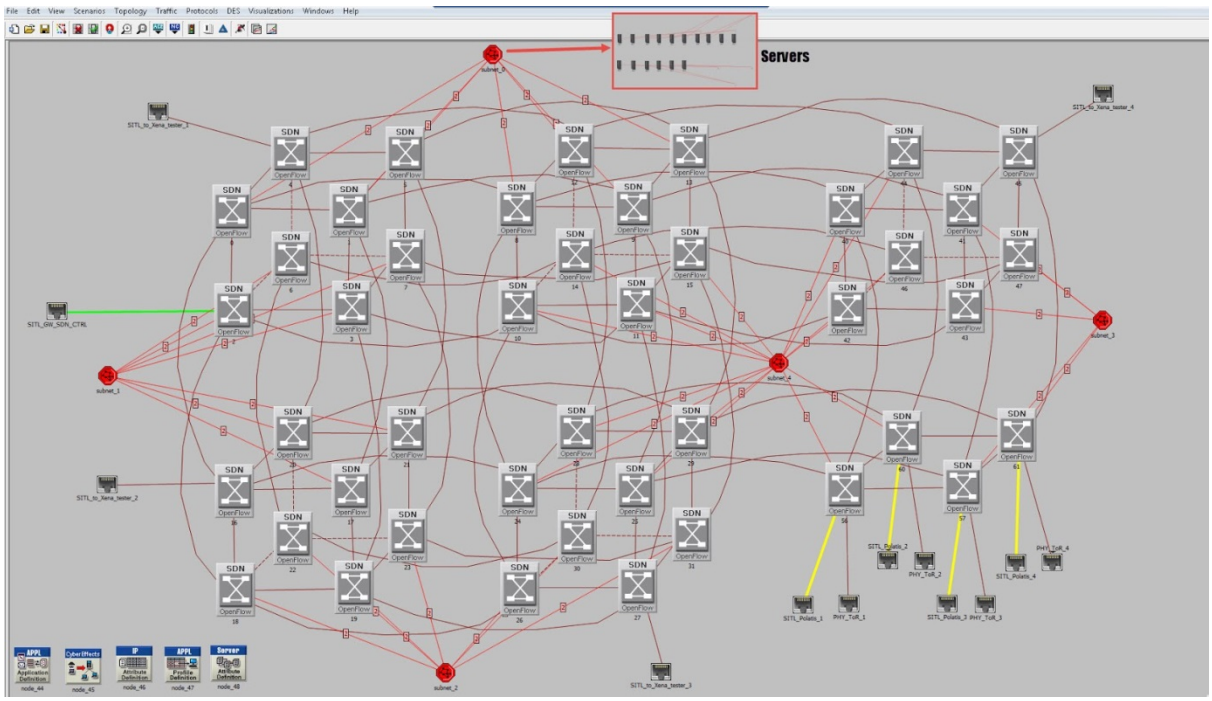


Figure 4-17: SITL model 3: 44-hypercube model + connectivity to Polatis + Xena testers

Summing up, the approach presented here, showcasing the combination of real hardware and simulation, has a large potential for the emerging integration of optics in the datacenter world, where scaling and latency effects must be studied without necessarily having access to real datacenter infrastructures.

4.1.3 Topology Visualization

In datacenters, it can be a big challenge to keep an overview of and to illustrate the connectivity of the servers and switches. Thus in COSIGN, we have chosen to use the Hyperglance [17] tool for topology visualization. The Hyperglance tool talks directly to the controller via a REST API, and interactively discovers connectivity and provides end-to-end topology maps. Furthermore, it allows 3D rotation of the topology, giving a complete overview from multiple angles. Each of the connected components contains related attributes and statistical data that can be accessed through a dashboard view. Figure 4-18 illustrates a 3D Hypercube in Hyperglance. In addition to the topology visualization capabilities (Open Flow-enabled network nodes), this platform allows performing dynamic data discovery and aggregation from multiple source platforms, including application container orchestration platforms (Docker and Kubernetes), virtualization tools, network state (alarms, failures) and performance monitoring frameworks. As a result, we can use this tool for the topological relationship mapping within and between different hardware and software platforms; hence, a more complex datacenter setup can be visualized in a multi-layer fashion, i.e., mapping functional dependencies at the service/application layer (e.g., a multi-tier web application, OpenStack resource deployment, Docker/Kubernetes application containers) as well as the network topology itself.

In addition, this visualization framework provides the following important capabilities:

- Graphical/color-coded state indication of the discovered network nodes (OpenFlow switches), links, interfaces.
- Support of the context-aware actions (direct remote access of the visualized end-devices via SSH, RDP, VNC, Telnet clients)
- Interactive control of the topology (flow installation, test link failures)
- Context-aware filtering of the visualized topology, e.g., filtering the components to display by type (network nodes, links, service/application components). Thus, a complex topology can be analyzed in a layered/modular way.

Figure 4-18 presents a 3D view of an 8-node Hypercube, consisting of 8 commercial HP Aruba SDN switches. Corresponding real-time configuration data such as installed flow records, protocol details, etc., are obtained from every tracked network device.

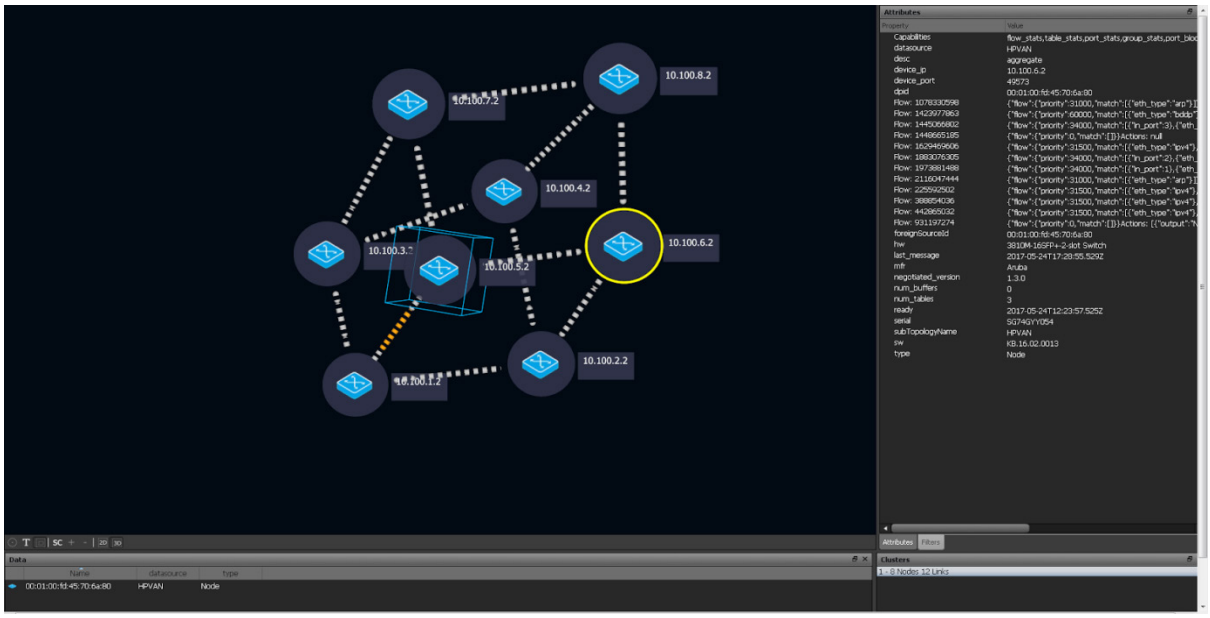


Figure 4-18: 8-hypercube model visualized in 3D via Hyperglance visualization framework

4.1.4 Summary

In this section the Hypercube structure with an optical shortcut has been presented as a proof of concept for datacentres. The optical shortcut facilitates load balancing, traffic prioritization and robustness towards failures. Latency measurements using Xena testers support the benefit of the optical shortcut. This however must be balanced with the reconfiguration time from the controller and how densely interconnected the optical shortcut switch is.

Using the Riverbed System in the Loop (SITL) tool, we have integrated simulation and real equipment. This allows us to study any datacenter architecture without possessing actual hardware in terms of performance and scalability, and to compare and integrate simulation models with real world hardware and software components. It will also give students great opportunity to conduct research with limited real world datacenter resources.

Finally, we introduced a 3D visualization of the datacenter dependencies, allowing to gain overview of complex datacenter structures.

4.2 Scenario 2: Cluster-based DCN

The intent for the Long Term Demonstrator is an experimental architecture for VDC networks. The intent is to explore the combination of optical TDM and circuit switching. In this section we will describe the final deployed demonstrator and describe the data plane performance characteristics as well as the development of the VDC use case that was extended from the mid-term demonstrator and successfully deployed on the long-term demonstrator.

4.2.1 Scenario

A description of the Long Term Demonstrator setup is given in D5.2 "*Demonstrator Result of Data Plane and SDN Environment Integration*" [4]. This includes details of software modules and integration efforts. We will include some detail here also, but [4] should also be consulted for more details. In figure 4-19 we illustrate the long-term demonstrator data plane. In D5.1 [16] we detail the use of HCF in the TDM data plane.

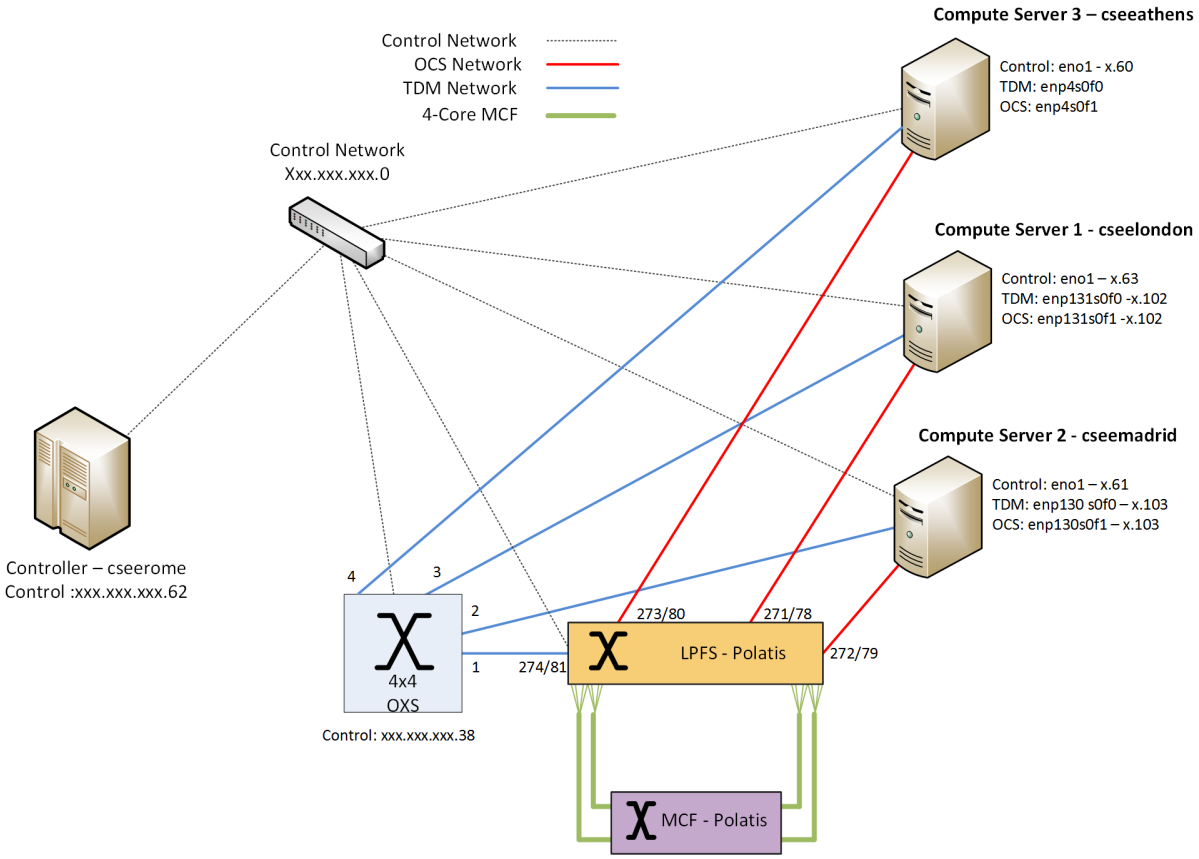


Figure 4-19 - The long-term demonstrator experimental configuration

The Intra-rack connectivity is provided by the TDM network, consisting of the plan-B OXS and TDM FPGA NIC cards. Inter-rack and high bandwidth connectivity is provided by the LPFS, with an MCF switch device intended to serve high-bandwidth inter-DC traffic.

4.2.2 Test plan and results

The tests employed on the long-term demonstrator are targeted at understanding the impact of the data plane behaviour on the application-level, for example the VDC use case. The behaviour of the data plane components has largely already been explored in previous research, but this is the first time they have been integrated and exploited by the application layer. Therefore, it was important to measure data plane performance from the perspective of applications.

4.2.2.1 Data plane performance testing

The performance of the circuit component of the data plane (the direct connections to the first layer LPFS) is well understood, providing a full 10Gb/s capacity and has been previously characterised and studied. Our testing was made up of the following components, some of which are presented in [16]:

- The TDM data plane was tested to establish throughput and latency characteristics for different TDM hardware configurations and software TDM slice (effectively bandwidth) allocations and allocation patterns.
- The effects of using Hollow Core Fibre in the TDM data plane were investigated.
- The application-level reconfiguration latencies of the TDM switch and MCF switches were established.

4.2.2.1.1 Switch Characteristics

The application layer switching characteristics of the data plane was obtained, by measuring the time taken for the application-layer network to become active when reconfiguring cross-connections. The hardware-level latencies were measured experimentally by the device creators and the results are shown in the following table.

	OXS Fast Switch	Polatis MCF Switch
Hardware latency	~ 25ns	< 25µs
Application layer latency	33µs	121µs

4.2.2.1.2 TDM Data Plane Behaviour

The performance of the TDM data plane was measured in terms of throughput and latency against allocated timeslots. These results demonstrate a sustainable maximum data rate of up to 8.6Gbps, see figure 4-20 and figure 4-21. As can be observed better latency and throughput can be achieved with interleaved (or distributed) slots allocations. This is because interleaving reduces the maximum delay, or number of timeslots between data transmissions, when less than the maximum number of slots has been allocated to a flow. It is therefore recommended to avoid contiguous allocation for best performance. Similarly, in figure 4-20 the maximum and mean latency measurements converge as the number of time-slices increases because the largest gap between active transmission slots reduces. The interleaved minimum is greater because unlike contiguous, there is always a no-transmit slot between transmissions. All results for the interleaved slot allocation are limited to 48 out of the 96 slots, allocating more than this and some slots are necessarily contiguous.

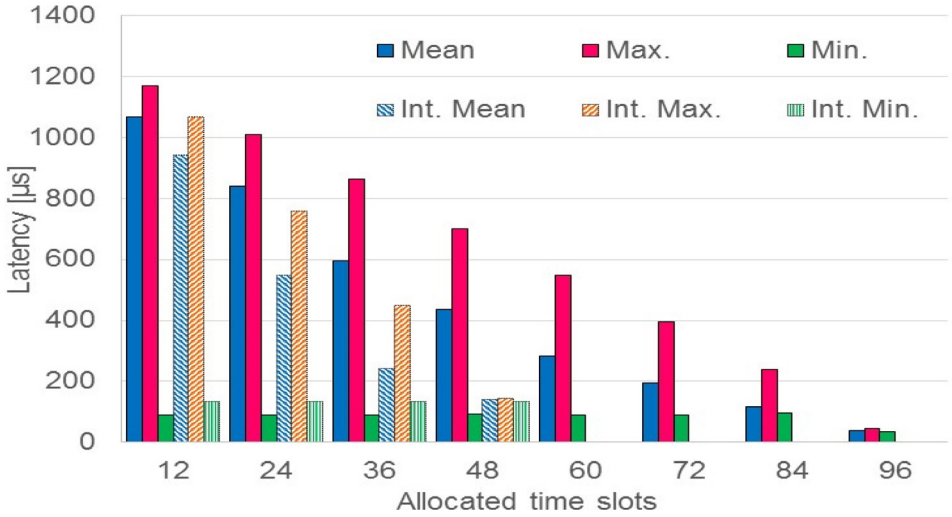


Figure 4-20 - Latency vs. number of allocated timeslots for contiguous and interleaved (Int.) timeslot allocations

In figure 4-21 we observe a larger throughput increase when we increase allocation from 84 to 96 timeslots than for any other step. This is because at 96 slots the entire TDM frame is not utilised and so the only gaps in transmission are the key characters used by the TDM scheme for negotiation.

Extra analysis of the TDM data plane is presented in [16] where we consider the impact of Hollow Core Band-Gap Fibre.

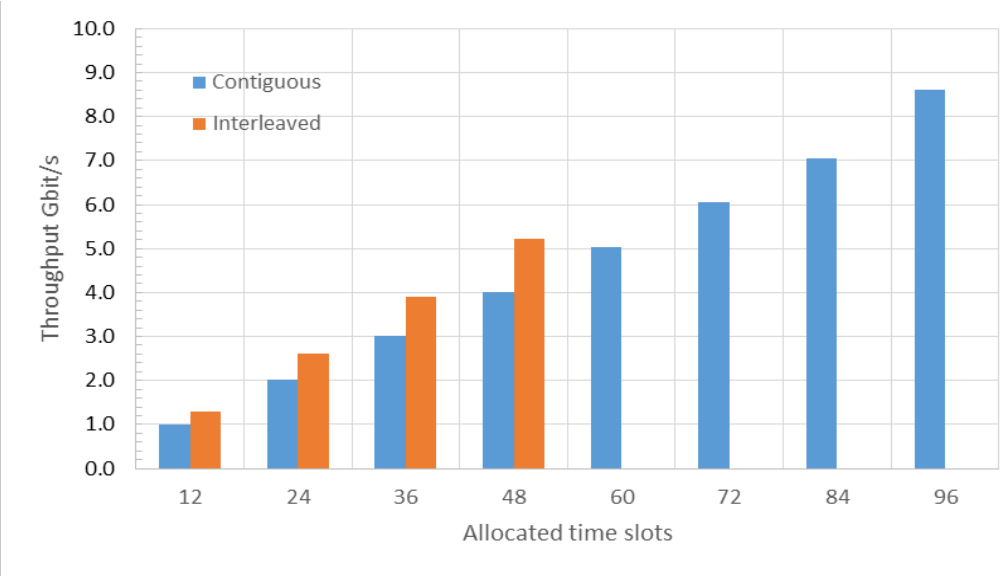


Figure 4-21 - Throughput for contiguous and interleaved allocated timeslots

4.2.2.2 VDC in the long-term data plane

As stated in [3], VDC provisioning in the long-term demonstrator must be able to utilise the TDM and OCS components depending on the requested bandwidth for a VDC instance. Extensions were made to the OpenFlow plugin and OpenFlow Java modules in ODL that enabled control of TDM NICs and the Plan B OXS device. Additional development of resource data models, path computation and provisioning in ODL was required to enable the provisioning of TDM resources via the Northbound REST interface.

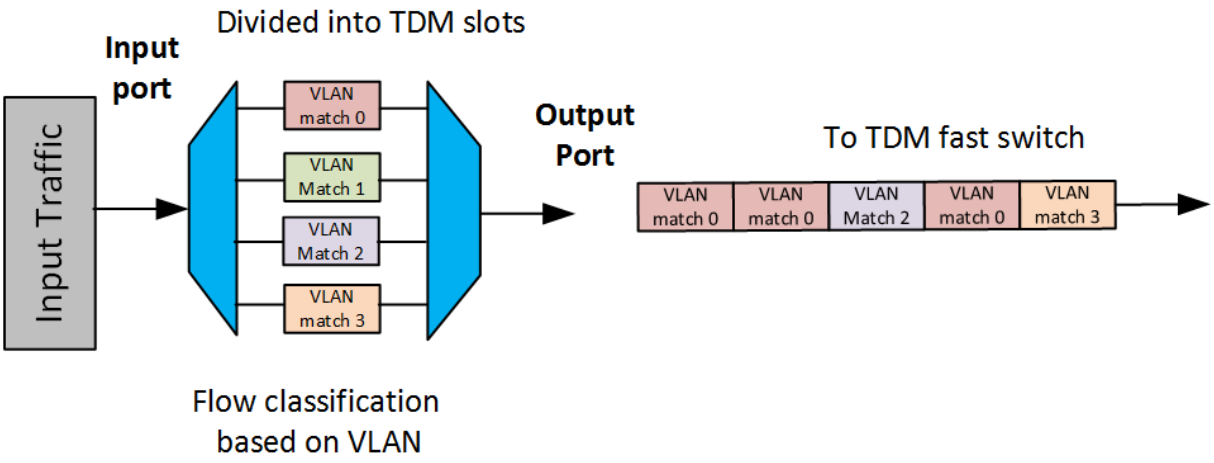


Figure 4-22 - TDM multiplexing on the FPGA TDM NIC based on VLAN ID

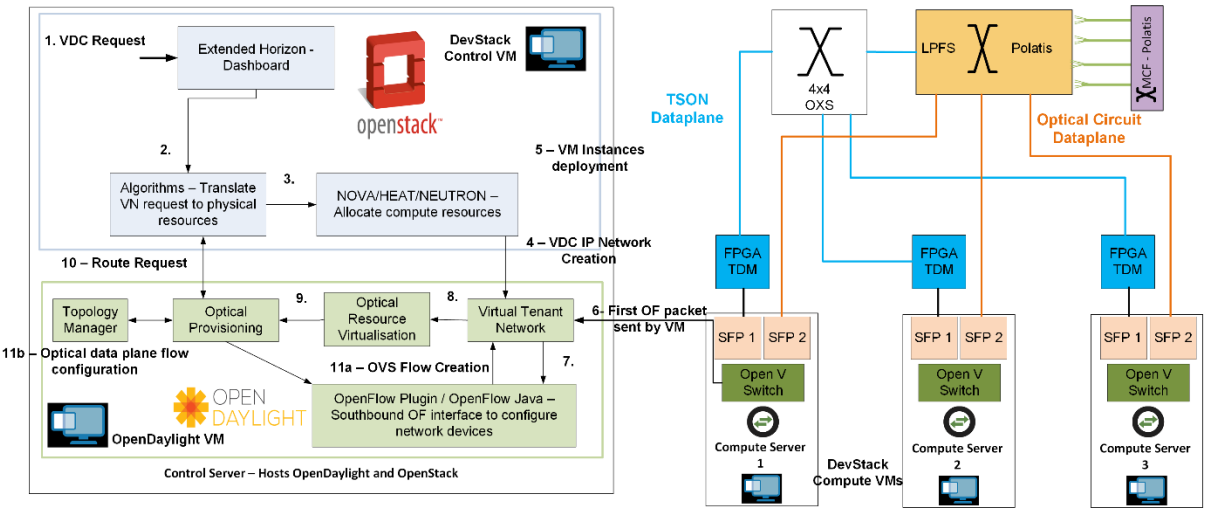


Figure 4-23 - Software control flow and hardware layout for thew Long Term Demonstrator VDC Use Case

This enabled OpenStack, in conjunction with an extended COSIGN *Algorithms* module to deploy VDCs onto the long-term demonstrator. The key upgrade to the algorithms modules was to enable selective allocation of TDM slices of complete circuits depending on the requested bandwidth. For the optimized provisioning of optical resources, a novel algorithm is used to translate tenants' bandwidth requirements into a request for TDM slots or an optical circuit, see Figure 4-22. The extended Northbound REST interface of ODL is used to interact with Optical Resource and Provisioning Modules to create the flow necessary to allocate the requested Virtual Network (VN). On each OpenStack compute node an OpenVswitch instance is programmed to control flows between VM instances. The utility of these instances is illustrated in Figure 4-23. Additionally, the new algorithms module developed specifically for this architecture determines the several logical instances (IP network, sub-network and ports) necessary to enable traffic exchange along the VDC instance, see Figure 4-24. To map the VMs and create the logical resources, it interacts with the core orchestrator services via the OpenStack Heat service. As well as the physical route and the necessary time slots, it also determines the particular VLAN to be employed when encapsulating the traffic of each virtual link, see figure 4-24.

These instances are used because the FPGA TDM NIC can multiplex the TDM slices, but can only match on VLAN ID, not source-destination MAC addresses as is desirable for a VDC network, see figure 4-22. In figure 4-24 we illustrate of the OVS instances on the compute node to translate the source-destination MAC addresses of communicating VMs (these VMs and their virtual network are provisioned in figure 4-25) into a VLAN ID which is then added to the flow's packets. This enables the TDM NIC to multiplex the incoming flows as determined by the orchestration layer. At the receiving side the VLAN ID can be stripped and the packets again forwarded by MAC address.

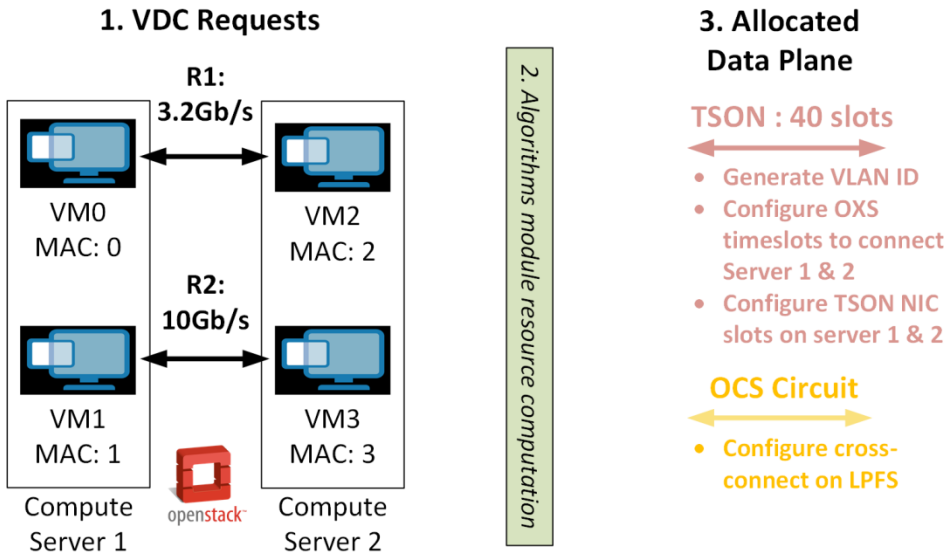


Figure 4-24 – Actions of the OpenVswitch instance on the OpenStack compute nodes to enable selective optical data plane usage

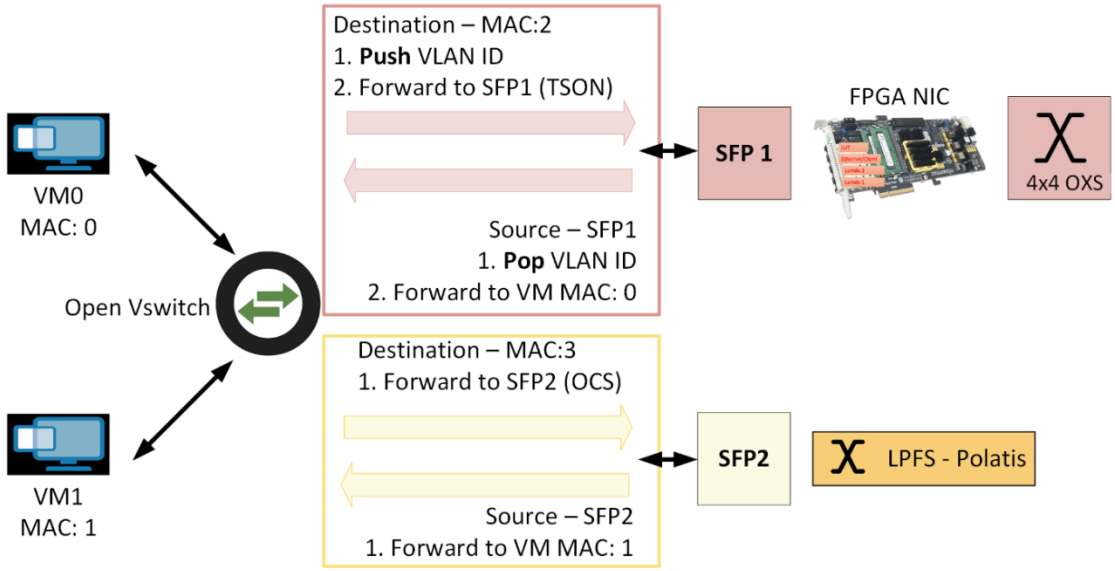


Figure 4-25 - Action of algorithms module to selectively allocate different data plane resources based on bandwidth request

5 Conclusions

The demonstrators in the COSIGN project have succeeded in demonstrating integration of various components developed in the project. Vertical integration of several layers of software on both control and orchestration layers have been successfully integrated with novel switches, fibres and network concepts developed in the project.

The integration and demonstration efforts in COSIGN have produced a number of key results reported in this document:

- COSIGN has demonstrated an up to 42% improvement in the number of accepted VDC instances on the same infrastructure when performing the developed joint provisioning of VMs and network configuration as opposed to legacy approaches with independent configuration. As confirmed by the industrial validation this enables a more efficient use of existing data center resources. This approach of joint optimisation is expected to play a significant role in future developments of data center control and orchestration mechanisms. COSIGN has produced and validated a substantial contribution to this development.

- COSIGN has demonstrated significant improvement (150% for the evaluated implementation) in throughput for selected elephant flows by applying the vApp traffic monitoring and data plane reconfiguration in an advanced circuit sharing scheme. The 150% improvement comes at the cost of only 37.5% reduction in throughput for the existing flow on the shared connection. This is a key result proving the benefit of adaptively reconfiguring data plane connectivity based on real time monitoring of data traffic. The improvement from optical circuit reconfiguration increases to even higher values when higher interface rates are connected to the optical switch. This result highlights one of the potential benefits of having optical circuit switches as part of the core of the DCN. As opposed to electrical switches, optical switches are not limited to a fixed total throughput capacity. There is thus no direct trade-off between interface rates and switch radix.

- COSIGN has demonstrated 50% improvement in round trip time (RTT) for mice flows when using circuit sharing according to the vApp use case. Additionally, the remaining mice flows benefit from the reconfiguration of the data plane reducing the load otherwise caused by the elephant flows.

- COSIGN has confirmed in an industrial setting that the developed VDC approach introduces negligible delay and complexity on the network side making it a highly attractive approach for industrial operators to investigate further. Key components developed in the project have been installed in an industrial setting and tested to provide valuable insight into the applicability in a commercial environment. The VDC provisioning approach developed in the project has been confirmed to offer significant benefits compared to currently used systems.

- COSIGN has provided a significant advance in the automated provisioning of VDCs as the COSIGN VDC approach can provide VDCs with guaranteed QoS with the same ease and speed on the part of the provider as “best-effort” VDCs are provided today. This is also confirmed in the industrial demonstrator and will potentially provide great value for operators adopting this approach. The provisioning speeds have been tested and confirmed to be low enough for a commercial environment. The capacity in terms of number of processed requests has been confirmed to be large enough for even large-scale operators.

COSIGN has also investigated the use of different network topologies to evolve DCNs while relying on data plane components from the mid-term demonstrator. Particularly the use of Hyper Cube topologies well known from the world of high-performance computing has been investigated. The combination of Hyper Cube topology with optical circuit switches is considered promising and substantial latency improvements are observed when using optical switching to reduce hop-count in

the network. In order to test topologies in large-scale networks a hybrid simulation - 'real network' platform is being established and proof of concept studies were conducted. Interfaces between the modelled network and the real network are being consolidated but preliminary results indicate that this is a very promising tool for performing reliable scaling studies which could not otherwise be carried out with only a limited access to DC equipment.

Apart from these immediately applicable results COSIGN has also investigated more disruptive developments of DCNs. As part of the long-term demonstrator an all-optical data plane has been implemented. A main achievement here has been to overcome the challenge of low granularity optical connections in a circuit-based network approach. This has been managed by introducing TDM-based circuits and demonstrating their compatibility with the developed control and orchestration tools. Consequently, the VDC provisioning scheme was successfully implemented on the all-optical data plane – capable of provisioning circuit connections based either on TDM- or full fibre connections. This is a completely novel approach to building an optical data plane and holds significant promise for when size and energy consumption prevents scaling using legacy approaches.

COSIGN has brought together key European partners in the area of DCN technologies covering both software and hardware. During the project, technologies from the different domains have been successfully integrated producing world-leading results both within each technology domain and through the comprehensive integration. There is no doubt that COSIGN has left its mark on international research in data center networks and has significantly benefitted European research and industry within the field.